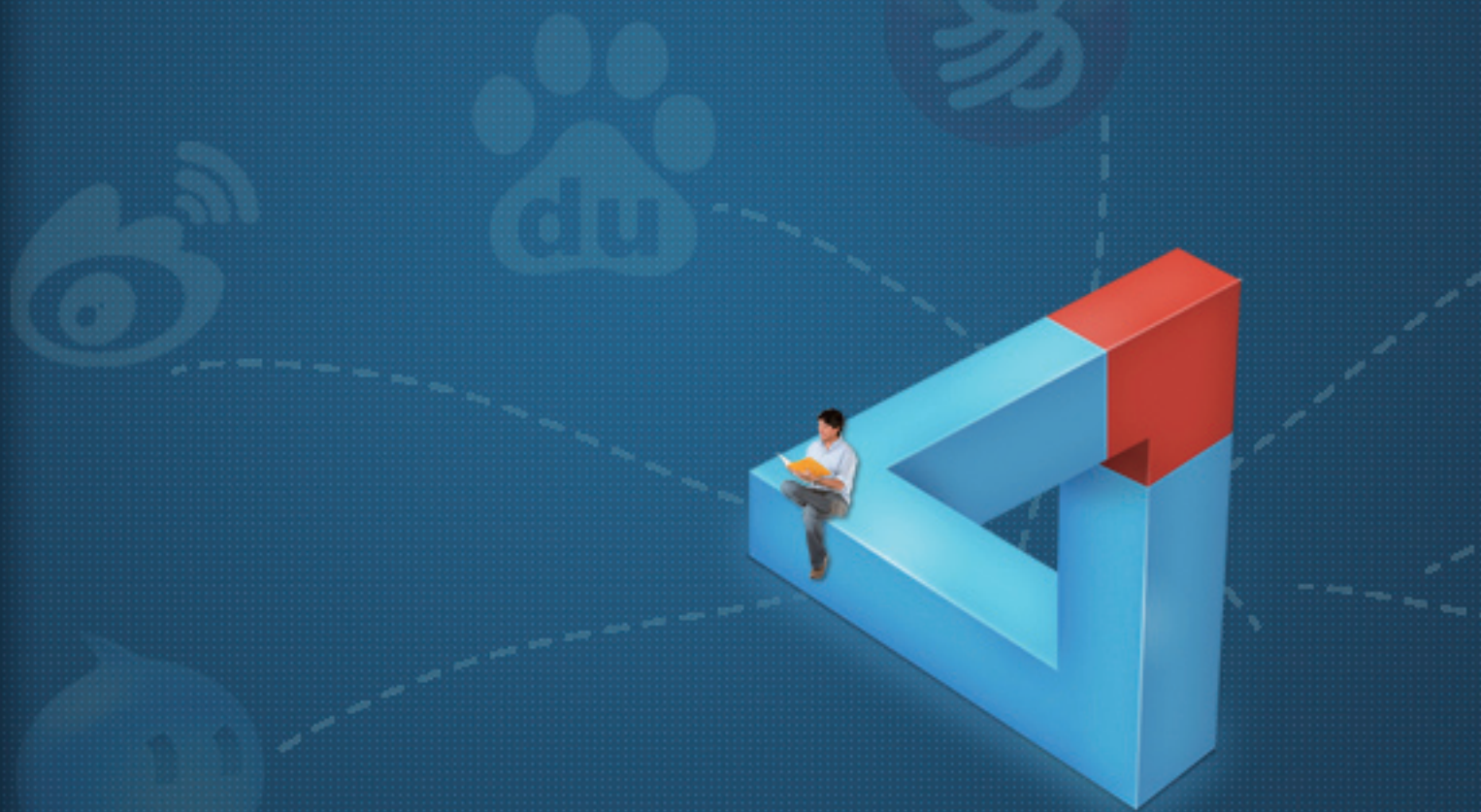


开发月刊

Development Monthly

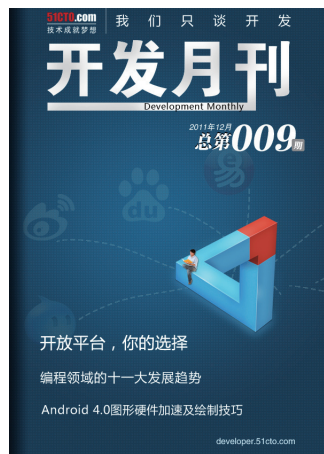
2011年12月
总第009期



开放平台，你的选择

编程领域的十一大发展趋势

Android 4.0图形硬件加速及绘制技巧



编程排行 Billboard

- 3 12月排行榜：C++11像一个新语言

专题报道 《开放平台，你的选择》

- 7 百度用搜索为开放平台注入活力
9 360开放平台以中立姿态迎接开发者
11 新浪微博平台围绕核心产品服务
12 人人网李福松：安全开放才健康长久

技术热点 Tech Focus

- 14 编程领域的十一大发展趋势
16 JavaScript 6看上去很美
17 什么情况该选择开发Web App?
19 jQuery开发技巧和心得
21 高效整洁CSS代码原则

移动开发 Mobile Dev

- 23 Android 4.0图形硬件加速及绘制技巧
28 与iOS的较量 Android 4.0功能评测
31 开发者必知：提升Android应用开发性能的十大要点
34 《Network World》主编谈“Lua编程语言”
36 手机不如家用机 移动游戏为何总倚老卖老?
38 马宁：越早进入Windows Phone开发越有优势
41 王涛：你必须知道的Windows Phone
44 李靖南：详解Windows Phone开发细节及注意事项

特别推荐 Best topic

- 47 51CTO专项调研:2011年度移动开发者生存状况

12月排行榜：C++11像一个新语言

2011年12月5日 Tiobe 发布了2011年12月的编程语言排行榜。新一期排行榜 Objective-C 持续给力跃居第五位,而上一期关注的谷歌 Dart 语言也上升到第68名,但 PHP 由第四位跌到第6名。

从2001年 Tiobe 编程语言排行榜开始发布至今, C++ 几乎一直占据着第三名。虽然 Perl, Visual Basic 和 PHP 曾经也排到第3名,但和 C++ 相比,他们就显得弱爆了,因为他们最多只能保持几个月而已。然而相比 C++, 微软的编程语言 C# 却更被看好,甚至有人认为 C# 势必会取代 C++, C++ 是垃圾语言。因为微软的大力推广,使用 C# 的人越来越多,而老将 C++ 却正在被人遗忘。不甘寂寞,2011年 C++ 再次亮剑,13年来第一个重大修订版, C++11 横空出世!

下面是前20名的编程语言排行

Position Dec 2011	Position Dec 2010	Delta in Position	Programming Language	Ratings Dec 2011	Delta Dec 2010	Status
1	1	=	Java	17.561%	-0.44%	A
2	2	=	C	17.057%	+0.98%	A
3	3	=	C++	8.252%	-0.76%	A
4	5	↑	C#	8.205%	+1.52%	A
5	8	↑↑↑	Objective-C	6.805%	+3.56%	A
6	4		PHP	6.001%	-1.51%	A
7	7	=	(Visual) Basic	4.757%	-0.36%	A
8	6		Python	3.492%	-2.99%	A
9	9	=	Perl	2.472%	+0.14%	A
10	12	↑↑	JavaScript	2.199%	+0.69%	A
11	11	=	Ruby	1.494%	-0.29%	A
12	10		Delphi/Object Pascal	1.245%	-0.93%	A
13	13	=	Lisp	1.175%	+0.11%	A
14	23	↑↑↑↑↑↑↑	PL/SQL	0.803%	+0.24%	A
15	14		Transact-SQL	0.746%	-0.03%	A
16	16	=	Pascal	0.734%	-0.03%	A
17	18	↑	Ada	0.632%	-0.02%	B
18	35	↑↑↑↑↑↑↑	Logo	0.619%	+0.26%	B
19	17		Assembly	0.563%	-0.10%	B
20	25	↑↑↑↑	ABAP	0.560%	+0.01%	B

掀起 C++ 11 的神秘面纱

C++ 标准在1998年获得通过后,有两位委员会委员预言,下一代 C++ 标准将“肯定”包括内置的垃圾回收器 (GC),但可能不会支持多线程,因为定义一个可移植的线程模型涉及到的技术太复杂了,13年后,新的 C++ 标准 C++11 也接近完成,你猜怎么着?让那两位委员没想到的是,本次更新还是没有包括 GC,但却包括了一个先进的线程库。C++ 之父 Bjarne Stroustrup 说 C++11 就像一个新语言,的确, C++11 核心已经发生了巨大的变化,它现在支持 Lambda 表达式,对象类型自动推断,统一的初始化语法,委托构造函数, deleted 和 defaulted 函数声明 nullptr,以及最重要的右值引用。

C++11 中值得关注的几大变化

- ◆ Lambda 表达式
- ◆ 自动类型推断和 decltype
- ◆ 统一初始化语法
- ◆ Deleted 和 Defaulted 函数
- ◆ nullptr
- ◆ 委托构造函数
- ◆ 右值引用

查看详细:

(<http://developer.51cto.com/art/201106/270597.htm>)

12 月编程语言排行榜: C++11 它就像一个新语言 II

C++11 标准库

C++ 于 2003 年以库技术报告 1(TR1) 的形式经历了重大改版, TR1 包括新的容器类 (unordered_set, unordered_map, unordered_multiset 和 unordered_multimap) 和多个支撑正则表达式、元组和函数对象封装器等的新库。随着 C++11 标准获得通过, TR1 和自它首次发布以来新增的库被正式纳入标准的 C++ 标准, 下面是 C++11 标准库的一些特性:

线程库

站在程序员的角度来看, C++11 最重要的新功能毫无疑问是并行操作, C++11 拥有一个代表执行线程的线程类, 在并行环境中用于同步, async() 函数模板启动并行任务, 为线程独特的数据声明 thread_local 存储类型。如果你想找 C++11 线程库的快速教程, 请阅读 Anthony William 的“C++0x 中更简单的多线程”。

新的智能指针类

C++98 只定义了一个智能指针类 auto_ptr, 它现在已经被废弃了, C++11 引入了新的智能指针类 shared_ptr 和最近添加的 unique_ptr, 两者都兼容其它标准库组件, 你可以在标准容器内安全保存这些智能指针, 并使用标准算法操作它们。

新的算法

C++11 标准库定义了新的算法模仿 all_of(), any_of() 和 none_of() 操作, 下面列出适用于 ispositive() 到 (first, first+n) 范围, 且使用 all_of(), any_of() and none_of() 检查范围的属性的谓词:

```
#include <algorithm>
//C++11 code
//are all of the elements positive?
```

```
all_of(first, first+n, ispositive()); //false
//is there at least one positive element?
any_of(first, first+n, ispositive()); //true
// are none of the elements positive?
none_of(first, first+n, ispositive()); //false
```

一种新型 copy_n 算法也可用了, 使用 copy_n() 函数, 复制一个包含 5 个元素的数组到另一个数组的代码如下:

```
#include
int source[5]={0,12,34,50,80};
int target[5];
//copy 5 elements from source to target
copy_n(source,5,target);
```

算法 iota() 创建了一个值顺序递增的范围, 好像分配一个初始值给 *first, 然后使用前缀 ++ 使值递增, 在下面的代码中, iota() 分配连续值 {10,11,12,13,14} 给数组 arr, 并将 { 'a', 'b', 'c' } 分配给 char 数组 c。

```
include <numeric>
int a[5]={0};
char c[3]={0};
iota(a, a+5, 10); //changes a to
{10,11,12,13,14}
iota(c, c+3, 'a'); //{'a','b','c'}
```

C++11 仍然缺乏一些有用的库, 如 XML API, 套接字, GUI, 反射以及前面提到的一个合适的自动垃圾回收器, 但 C++11 的确也带来了许多新特性, 让 C++ 变得更加安全, 高效, 易学易用。

如果 C++11 的变化对你来说太大的话, 也不要惊慌, 多花些时间逐渐消化这一切, 当你完全吸收了 C++11 的变化后, 你可能就会同意 Stroustrup 的说法: C++11 感觉就像一个新语言, 一个更好的新语言。

Lambda 表达式

Lambda 表达式允许你在本地定义函数, 即在调用的地方定义, 从而消除函数对象产生的许多

12 月编程语言排行榜: C++11 它就像一个新语言 III

安全风险, Lambda 表达式的格式如下:

```
[capture](parameters)->return-type {body}
```

[] 里是函数调用的参数列表, 表示一个 Lambda 表达式的开始, 让我们来看一个 Lambda 例子:

假设你想计算某个字符串包含多少个大写字母, 使用 `for_each()` 遍历一个 `char` 数组, 下面的 Lambda 表达式确定每个字母是否是大写字母, 每当它发现一个大写字母, Lambda 表达式给 `Uppercase` 加 1, `Uppercase` 是定义在 Lambda 表达式外的一个变量:

```
int main()
{
    char s[]="Hello World!";
    int Uppercase = 0; //modified by the
lambda
    for_each(s, s+sizeof(s), [&Uppercase]
(char c) {
        if (isupper(c))
            Uppercase++;
    });
    cout<< Uppercase<<" uppercase
letters in: "<< s<<endl;
}
```

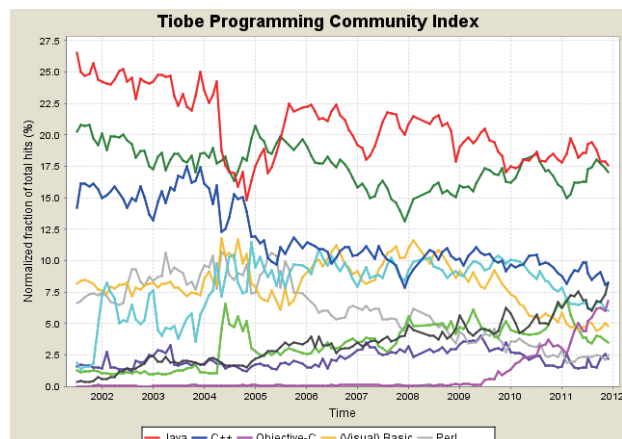
这是因为, 如果你定义的函数主体被放置在另一个函数调用内部, `[&Uppercase]` 中的 “&” 记号意味着 Lambda 主体获得一个 `Uppercase` 的引用, 以便它能修改, 如果没有这个特殊记号, `Uppercase` 将通过值传递, C++11 Lambda 表达式也包括成员函数构造器。

(未完请查看网络原文:

<http://developer.51cto.com/art/201112/305880.htm>

下面是本期编程语言排行榜的其他排名数据和趋势走向。

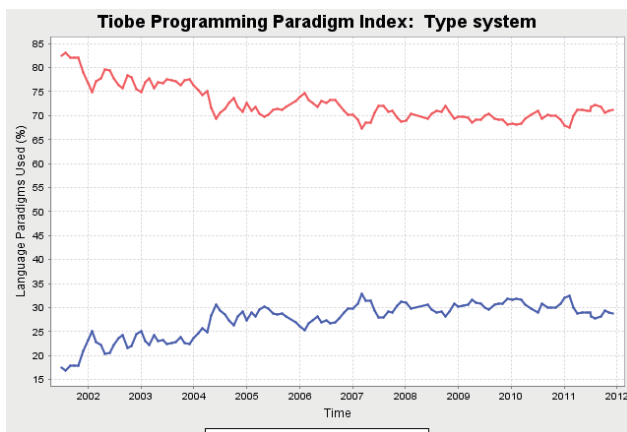
前 10 名编程语言走势图



下面是第 50 到 100 的编程语言排名

(Visual) FoxPro, ABC, ActionScript, Algol, Alice, Bash, bc, BETA, BlitzMax, Boo, Bourne shell, CFML, cg, CL (OS/400), Clean, Clojure, cT, Dart, Dylan, Eiffel, Euphoria, Factor, Icon, IDL, Io, J, JavaFX Script, JScript.NET, Korn shell, Lingo, MAD, MUMPS, NATURAL, Oberon, Occam, OpenCL, OpenEdge ABL, Oz, PowerShell, REXX, S, Scala, SPSS, Standard ML, SuperCollider, VBScript, Verilog, X10, xBase, Z shell

下面给出了编程语言类别的一年变化趋势



开放平台 ● 你的选择

中国软件开放平台众多，作为开发者的你如何选择？51CTO推出《开放平台，你的选择？》年终策划，集中展示国内目前最具影响力的几大开放平台。内容涵盖平台优势、优秀案例、开发者利益分成模式、授权机制和安全保护措施等方面。让各位开发者在作出自己选择时，更加踏实少走弯路。



51CTO 开发频道 2011 年终巨献

中国软件开放平台众多，作为开发者的你如何选择？51CTO 推出《开放平台，你的选择？》年终策划，集中展示国内目前最具影响力的几大开放平台。内容涵盖平台优势、优秀案例、开发者利益分成模式、授权机制和安全保护措施等方面。让各位开发者在作出自己选择时，更加踏实少走弯路。

更多内容请查看“开放平台，你的选择”专题：

<http://developer.51cto.com/developer/platform/index2011.html>

51CTO 开发频道寄语

百度用搜索为开放平台注入活力

望金蓉: 百度开放平台产品经理。2010年6月开始参与百度应用开放平台的研发工作,现全面负责百度应用开放平台的产品和运营。很多人将百度“框计算”视作咱们常见的输入框,在51CTO看来这个“框”其实更像西部牛仔手中的套索,准确套中用户搜索需求。



总体来说,百度应用开放平台服务的开发者不分行业,也不分团队或公司大小,我们致力为开发者打造一个能通过搜索响应网民需求的开放平台。只要用户有需求,而开发者又能为用户带来优质的体验,这个领域就可以无限扩展。从用户需求角度,我们可以将平台上承载的应用分为游戏、视频、音乐、阅读、工具、生活服务等几大类。在这些分类作品中表现突出的,既有像POPCAP(植物大战僵尸)这样知名的国际游戏公司,也有像4399和7k7k这样的国内小游戏产业典范。而更多的是个人开发者和创业团队,比如阿达软件、问阅堂工作室、贝瓦网等。他们各有特色的应用作品覆盖各行业各领域。

针对不同类型的应用,百度应用开放平台提供有针对性的对接策略和服务方式。比如视频、图书(阅读)、音乐的版权资源主要集中在大公司或代理方平台上,我们会提供批量对接、测试工具和上线机制,方便资源合作。以阅读类为例,百度提供封装工具和阅读器,合作方只需要提供txt等格式的文档,就可以自动转化成符合上线要求的阅读类应用。合作方无需开发,大大降低了合作门槛和成本。

为了更好地服务个人开发者或创业团队,百度应用开放平台可以提供应用部署环境和最高1TB的云存储空间,帮助他们解决域名、机器的问

题,大大节省创业成本。另外,我们还提供有“绿色通道”,一对一地帮助优秀开发者开发、提交和测试应用。

对于开发者而言,好的开放平台需要既帮助他们解决起步阶段的短期问题,也要解决长期需求,让开发者充分发挥其创造性。百度开放平台的特色以及保证开发者收益的策略参见“平台优势”(app.baidu.com/developer)和百度应用成长基金计划。

除了提供开放平台和激励策略,推动更多优质的应用资源与用户需求直接对接。百度还为开发者提供产品设计、开发方法方面的延伸服务,提供技术资料,引导开发者思考产品定位,交流技术心得。例如我们在今年举办的Web App应用开发者大会、Web App应用创新大赛以及开发者俱乐部系列活动。

百度应用开放平台的特点

1. 开放性: 搜索引擎作为网民上网的主要入口,其平台面向所有网民敞开,用户可以通过搜索引擎无门槛地接入和使用第三方应用,并快速形成使用习惯。

2. 对接海量需求: 百度每天响应数十亿次的搜索请求,让应用提供商与海量需求实现有效对接。

3. 需求精准匹配: 通过关键字实现需求对

百度用搜索为开放平台注入活力 II

接,开发者更精准地找到自己想要用户,用户也更精准地找到最优应用资源。

4. 优质资源:通过百度应用开放平台,更多优质的第三方应用和内容资源,将可以直接与用户需求无缝对接。

5. 多元化:现阶段百度应用开放平台上的应用,直接在浏览器中运行。未来会有涵盖pc、无线平台的应用,并且既有浏览器应用,也有独立应用。百度应用开放平台是一个面向所有合作伙伴(包括个人开发者)完全开放的平台。任何接入合作并通过审核的优质应用,都可以通过百度搜索与用户需求直接对接。

对于优质应用在百度搜索中的排名,以及同质应用资源之间的排序:

是以百度搜索的系统算法为基础;

结合用户对应用体验的满足感、需求热度、增长速度和幅度、口碑点评、粉丝数量等综合的欢迎程度和个性化定制(用户可收藏喜欢的应用,并定制到特定的词下展现);

参考应用本身上线的时间先后、稳定性、创新度、版本级别和优劣、以及后期更新升级等因素,最终根据用户输入需求的相关性,由百度搜索系统计算而产生。

总体而言,用户体验是基础,越受用户欢迎的应用,排名排序也就越靠前越稳定。当然,对于后上线的新优质应用,会根据同质或同类应用的相关性给予推荐,同时结合其上线后的用户体验增速和增幅情况,由系统自动对其排名排序进行相应的提升。

对于开发者权益的保护我们分为几个层面来进行。首先,百度应用开放平台要求开发者提

交的资源是合法,无版权纠纷的,并且有审核上线的流程。同时,我们更加鼓励开发者原创。以阅读类应用为例,作者和版权方都会得到显著体现,参见纵横中文网的《永生》等作品的展现效果。

百度应用开放平台最大限度地保障开发者权益。如果开发者使用百度提供的 OpenID,则授权与否取决于使用者的选择。

在用户使用百度账号登录的情况下,百度应用开放平台上的应用加载时,平台只会传递当前登录用户的 UID 信息给第三方。除此之外,平台不会主动给第三方传递任何用户数据,所有数据都由第三方通过 Open API 获取。Open API 的安全机制与业界主流平台的安全机制基本一致,自2011年6月开始采取 Https + access token 机制来保障安全。https 解决了数据传输的安全性问题,access token 解决应用对数据的访问权限问题,而授权则通过业界最为安全先进的 oauth2.0 协议。

框计算的精髓是高度智能的互联网需求交互模式,“最简单可依赖”的信息交互实现机制与过程。在框计算支持下,用户需求实现过程如下:

首先,用户任意一个需求被提交到“框”里;

然后,“框计算”经过一系列复杂的需求分析,包括语义分析、行为分析、智能人机交互技术分析和海量计算,将用户的需求分发给“框计算”后台单个或多个对应的数据/应用所响应;

第三,“框计算”背后的资源平台是开放的,框计算平台提供了大量即插即用的接口,各种数据和应用可以主动与框计算平台对接,使自己有机会来响应框所收集到的需求。 ■

文章未完,更多内容请参考以下链接:

<http://developer.51cto.com/art/201112/306332.htm>

360开放平台以中立姿态迎接开发者

石晓虹：奇虎公司副总裁、首席技术官。主要负责奇虎的搜索产品、战略合作和客户端软件业务。360 希望整个 360 开放平台能以中立平台的角度看待开发者，善待每一个应用开发人员。



总体来说，百度应用开放平台服务的开发者不分行业，也不分团队或公司大小，我们致力为开发者打造一个能通过搜索响应网民需求的开放平台。只要用户有需求，而开发者又能为用户带来优质的体验，这个领域就可以无限扩展。从用户需求角度，我们可以将平台上承载的应用分为游戏、视频、音乐、阅读、工具、生活服务等几大类。在这些分类作品中表现突出的，既有像 POPCAP(植物大战僵尸)这样知名的国际游戏公司，也有像 4399 和 7k7k 这样的国内小游戏产业典范。而更多的是个人开发者和创业团队，比如阿达软件、问阅堂工作室、贝瓦网等。他们各有特色的应用作品覆盖各行业各领域。

针对不同类型的应与其他开放平台相比，360 开放平台的差异化在于强化了软件、应用、游戏、移动互联网等多领域的开放；此外，360 更倾向于能以中立平台的角度看待开发者，应用数量也是目前同类领域中最多的。另外，基于平台的安全性，360 开放平台拥有天生的安全基因。

针对开放平台行业的未来发展趋势，360 认为，未来开放平台领域将更倾向于全平台的综合性发展，而 360 未来也将加强对开发者提供更多技术支持。

此前曾有业内资深人士认为，国内开放平台对开发者“区别对待”的现象非常严重，很多开放

平台只愿意对认可了“独家”和“投资”等合作方式的应用项目进行推广，这一行业现状令不少开发者感到无奈，存在竞争关系的同业厂商更是对此深恶痛绝，甚至将这类“开放”称为“伪开放”。

360 认为，“伪开放”的出现无形间对开发者形成了深度的伤害，相比之下，360 开放平台一贯以完全开放的思路来操作，由此生成的 15 万应用数量便是最好证明。

通过在开放平台领域发力，360 希望能给开发者提供一个更好的平台，借助在流量与技术方面的支持，帮助开发者建造一个共赢合作的生态圈。

开发者最关注的莫过于应用利益的分成和平台对应用的推广作用，360 开放平台对开发者应用流量的提升效果是非常明显的，未来将逐渐完善利益分成制度。

同时，360 已经针对开发者设立了 1 亿元的开发者奖励基金与 10 亿元的创新基金，以奖励和支持优秀的开发者和开发团队。

目前 360 开放平台授权分为几个部分：

① 内容类，包含视频、游戏、小说、杂志、音乐等，与业内的合作伙伴签订合作协议，只接受这些合作伙伴提供的内容，合作伙伴承诺保证版权。

② 应用类，开发者在开放平台注册时要求上传营业执照方能通过资格审核，遇到涉及版权、资

360 以中立者姿态迎接开发者 II

质的相关应用,都要求开发者提供相关的资质证明。审核通过后,才可以上线。

③ 我们设立了投诉电话、邮箱和举报平台,并有专人处理。在 360 桌面等产品上也设计了明显的举报入口,用户或机构可通过电话、邮件、举报平台等渠道反馈版权相关的问题。一旦接到举报,我们会第一时间将涉及的应用下线,并展开调查。如果证实应用开发者侵权,不会再接受其提交的应用。

360 开放平台目前采用多层架构设计,其中应用开发者只需要关注 360 开放平台的接入接口的功能和数据描述,关于如何与接口交互可以直接使用我们提供的 SDK (软件开发工具)来完成,非常简单。关于平台的服务器架构部署情况,简单的说我们对服务也采用了分层架构,在各层都实现了 failover 机制,可以保证服务的高可用性,同时我们的架构都是可以横向扩容的,可以保证对性能有高可扩展性。

比如其中最前端负载均衡服务我们主要是采用我们自己改造的 LVS 服务(加强了我们自己特有的功能)、WEB 业务服务器实现了自己的 failover 和监控报警方案和负载均衡、中间数据层(如 redis 集群)也有 failover 方案和和负载均衡并可以做到自动切换主从部署、MySQL 服务同时也实现了 failover 方案和负载均衡。也就是说我们在主要服务器的各层服务都做了高可用性的设计和部署。

在数据存储方面,360 开放平台使用了 MySQL+Redis+Cassandra 的组合,其中部分功能还使用了 MySQL+memcached+ TTServer (TokyoTyrant + TokyoCabinet)的方案。目前的

技术架构已经使用到了 NoSQL 数据库。

平台安全性至关重要,360 开放平台对于安全性的关注不仅在平台的安全性上,我们还关注用户使用的安全性,以及合作伙伴应用的安全性。

360 开放平台的应用接入采用业内标准协议 OAuth 做为用户授权应用的基本协议,其中 OAuth1.0a 使用签名加密和防重放来实现传输的安全性, OAuth2.0 使用 ssl 技术来保障传输的安全性。为了对用户隐私进行更好的保护,360 对应用接入有等级要求,部分高级合作功能只有有实力的合作授权级别才可以调用,此部分合作服务商都需要与 360 签署严格的协议。由于 360 的平台既有互联网站点、又有客户端,我们在客户端的应用接入上独创了基于 OAuth 的改进协议,保证客户端接入的安全。另外,对于接入我们开放平台的软件和手机应用,我们都会进行严格的安全检测、通过全球数十款知名杀毒软件的联合查杀才可以发布到我们平台,从而保证用户下载的软件和手机应用安全无毒。■

51CTO 精品杂志推荐《Linux 运维趋势》



邮件订阅入口:

<http://os.51cto.com/art/201011/233915.htm>

投稿邮箱: yangsai#51cto.com

发布周期: 每个月的第二个星期五

往期《Linux 运维趋势》下载汇总页:

<http://down.51cto.com/zt/71>

新浪微博平台围绕核心产品服务

王明功：新浪微博开放平台高级产品经理(个人微博：<http://weibo.com/minggong>)。负责开放平台 API 服务体系、开发者支持体系的架构及设计等。为网站、应用等接入微博平台，提供深入的技术、产品解决方案。



新浪微博开放平台提供安全的授权方式，稳定的接口调用，优雅接口设计，方便的应用接入以及优惠的云托管服务。开放平台最重要的是要围绕核心产品提供对外服务，并由此打造生态链。开放平台的特色、规模、盈利模式等都会因为核心产品的不同会有所不同，除此外围绕开放平台进行第三方的引导、扶持等也至关重要。

在中国现在的市场情况下，将会出现不同类型、不同层级的平台化服务，由于用户需求会有重合，平台以及开发者也会有重合。比如游戏，在无线平台、web 平台都有需求。

根据用户的需求和开发者需求以及平台的生态安全来共同设定，在考虑到用户隐私等问题的基础上，最大限度的开放微博平台内容。目前在用户授权的情况下，开放用户信息、关系信息、微博内容等核心服务，也提供了话题、地理信息等扩展内容，未来也将会提供更多内容，方便开发者通过开放的服务为用户提供更丰富的应用。

目前的扶植包括很多方面，针对游戏开发者我们推出了 1 年无分成的合作模式，在这一年内将游戏的所有收入都返还给开发者，目前不少游戏的盈利还是相当可观的，当然在产品功能上我们也提供了游戏中心，方便游戏进行多维度的内容分享，以更快更丰富的形式进行传播。对于普通应用，我们也推出了低价的云服务，开发者无需

考虑服务器托管问题，直接在我们的云平台上发布即可，站内应用每个月都会获得相应的云豆发放，几乎在零成本的情况下开发应用，同时也能分享微博的 2.5 亿用户，未来我们也会为各类开发者提供支付渠道帮助开发者盈利。

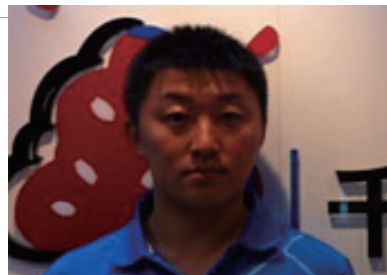
微博平台已经上线微币支付功能，目前已经提供给平台上接入的游戏使用，开放的第一年全额付给游戏开发者，未来将会以 7:3 的方式分成。针对普通应用，我们也在计划开放微币支付功能，将会以 7:3 的方式分成。在应用的用户传播上，我们也在不断探索多种方式，尽力在保证用户体验的前提下让应用最大化的传播，提升应用总体访问量，带来更多增值可能。

目前以 MySQL 数据库为主，在合适场景会考虑用 NoSQL，目前 NoSQL 方面主要使用了 Redis，也取得了一些不错的效果。

平台的开放是一把双刃剑，能够引入很多产品来丰富核心服务，也会带来各种对开放安全性的挑战，平台也是在不断完善中走向成熟。目前，微博开放平台提供了多维度的安全性规则，保障用户的安全以及为不同的开发者提供所需的权限。我们有强大的产品、技术、运营人员组成的团队来规划平台的安全机制，从接口设计、接口上线到接口开放、用户授权、接口监控，每一步都会有规范的流程确保。 ■

人人网李福松：安全开放才健康长久

李福松 人人网开放平台之父 高级产品 & 技术经理。在其他开放平台刚刚萌芽的时候,人人网开放平台就已经风生水起了。人人网早在2008年7月8日正式对外发布了开放平台战略,从此全面拉开了中国互联网的开放平台时代。开放平台成为创新者和创业者的乐土。



开放必须安全的开放才能保持长久健康的开放状态,因此对于安全验证及授权这块就尤为重要,人人网开放平台率先在国内采用了 OAuth2.0 的验收与授权解决方案,同时首先翻译了 OAuth2.0 的中文版贡献给国内社区,目前 OAuth2.0 支持多种的验证流程,在人人网开放平台接入应用方面发挥重要作用。

用户隐私是人人网一直非常看重的,而人人网的用户模型与其他网站有很大不同,因此我们在权限的粒度设计上充分照顾到用户的不同隐私需求,将权限的范围进行了详细的划分。

我们构建自己的通用开放平台 (Common Programming Platform, 简称 CPP), 目的在于增强人人网开放平台的开放性和标准化,从而使第三方与人人网的集成更加容易和多样化,进一步推动开放平台领域的社会分工,为产生各种开发 SDK 创造必要条件。

人人网开放平台在对外公布 REST API 的基础上,又对外发布了我们的 Widget API,将可视化的组件也通过 API 的定义方式对外发布,充实了我们对 API 的理解,这样的设计是我们在标准化的努力实践。

人人网开放平台努力封装各种语言的 SDK, 以及为 Discuz 等开源社区封装组件,就是力图降低开发者的接入门槛,同时 SDK 上不仅仅是简单

的接口封装,也逐渐将业务功能进行整合封装到 SDK 中。

开放平台一直秉承开放共赢的理念为客户服务,不为开发者设置过多门槛,同时为了平台的健康发展以及开发者群体的整体利益对于应用接入的产品质量严格把关,努力将精品推送给人人网用户。任何一个组织或个人,都可以自由的制作人人网应用,我们有专门的营运团队为开发者解答业务层面的问题,我们也有专门的技术咨询与支持团队为开发者进行技术支持,同时我们平台的运维部门还为那些优秀但缺少服务器的应用准备了一些服务器供开发者使用,目前这种服务器及带宽的租赁是免费的。同时我们在网站也有不同的资源对应用进行推广。

目前我们在执行的分成策略是,在扣除人人网渠道成本后的部分进行 4:6 分成,开发者占 6 成。同时开放平台也正在积极研究新的更加优惠的分成策略,让开发者得到更多利润用以鼓励开发者在人人网研发更多新的产品。

分成策略只是我们吸引开发者入驻的因素之一,我们通过与开发者广泛沟通发现能够给其带来大量流量,提升品牌价值才是开发者最为看重的,目前的开发者们更看重的是其团队的长远发展及品牌目标,因此在这方面人人网拥有得天独厚的优势可以帮助开发者们实现这一目标。

人人网李福松：安全开放才健康长久 II

除了分成策略,我们在自助广告的投放也向开发者的应用倾斜,我们在运维资源的支持上也有相对灵活与宽松的政策,很多开发者得到了我们大量服务器及带宽的免费支持。这些都从不同侧面保证开发者的合理利益得到保障。

通过 OAuth2.0 的发布,我们在“用户授权”这一部分,将平台提供的服务 (Service) 进行了标准化 (同时也是行业现成标准)。由于 OAuth 2.0 带来的标准化和实现的通用性,各种 Connect 客户端 (即 SDK) 就能交给开源社区来封装。通常,这样的开发工作只需要个人开发者就可以完成。

在这个基础上更进一步,在“用户授权”以外的领域,可以将整个平台各种服务进行标准化,打造出一个 Service 层 + 一套相关规范 (spec)。这样的技术架构,在“用户授权”以外的领域,尤其有利于 widget 产品 (暂且叫这个名字) 的标准化。关于这种通用 widget 的雏形,可以参考 facebook 的 Platform Dialog 和各种 social plugin。

我们从平台的各种接口中剥离出一个 Service 层,在 Service 层基础上,产生各种 SDK。这个 Service 层,加上相应的一套编程规范,组成整个开放平台的核心——通用编程平台 (CPP)。

平台的核心,通过标准化搭建起一个 Service Layer。这一层的工作必须由开放平台完成。

上层的两个 SDK 层次可由开源社区来完成。

第三方应用开发者根据需要可以选择使用三个层次的 API。

平台的核心——CPP 包括两部分:

(1) CPP Service Layer, 由各种 endpoint 组成。

(2) 由文档组成的 Spec, 包含 endpoint 需要的参数、回传方式、显示模式、注册设置等规范。

Service 层中的各个 endpoint 大部分是基于 HTTP 的一个服务点,但 SSO Endpoint 不是。它是某些在运行环境下进程间通信的接收和验证服务,其对应的 Spec 中包含进程间通信的地址、参数、安全验证设置等信息。

Service 层中的 endpoint 可以分为四大类:

- ◆ Auth Endpoints
- ◆ Renren API Endpoints
- ◆ Widget Endpoints
- ◆ Single Sign-on Endpoints

更全面说开放平台在数据存储上采用 DB+Cache (远程 + 本地) 的组合方式。

在 DataBase 方面采用通用的 Master/Slave 的模式,但也同时根据不同业务进行了垂直部署与水平拆分,不同的业务会由不同的 DB 集群承担,一个业务中数据量过大的会进行散表水平拆分;

有专门的 Hadoop 集群用来集中处理平台的各种 Log;

有专门的 Memcached 集群维护业务数据;

也对部份业务数据进行了本地化缓存。

人人网也在研讨 NoSQL, 开放平台在这方面也有跟进,也在时机成熟的时候大规模引入 NoSQL。

一方面我们采用安全程度高的授权与验证解决方案 OAuth2.0。同时我们在业务实现上尽量用简单实用的流程来降低负责度与提高安全性。

一方面我们通过多维的数据监控及时的发现异常变化。一方面我们与人人网安全部门紧密的合作,及时更新各种新发现的安全漏洞。我们有专门的人员规划平台的安全机制。 ■

编程领域的十一大发展趋势

核子可乐 / 译

根据我们各自的视角及对具体问题的切身感受,编程世界的发展到底是过快还是过慢在不同人的眼中会有不同的结论。但无论大家的关注重心是 Cobol 应用还是 Node.js 破解,有一项事实没人能够否认,即:程序员必须始终紧跟不断发展变化着的编程技术趋势,否则就无法保持自己在这一活跃领域中的竞争力。

从随处可见的 JavaScript 到无所不包的 JVM,时代与主流工具始终处于变化之中。程序员们的工作方式同样如此,多亏了框架与相关规范的出现,目前的发展方向开始摆脱一味追求开放性的误区。对于带宽、能耗以及可扩展性的关注使更多人开始从编程层面寻求解决方案,例如并行计算及显卡。这一切的一切都在我们身边迅猛发生,使得很多人对变化产生了恐惧感,甚至希望可以回到学校重新进修——只要那些传统的教学内容还能帮得上忙。

每隔一段时间,我们就要对进行编程工作所需要的方法及手段加以更新或是全盘替换,这有点像使用新的操作系统或是函数库。在本文中,我们将与大家分析十一大近来正在如火如荼发展着的技术趋势,借以共同把握编程领域的未来走向。

编程发展趋势第一位:JVM 不再只为 Java 服务

在很久很久以前,Sun 公司打造了 Java 并为全世界贡献了虚拟机这一宝贵财富。在微软推出 C# 语言的时候,人们意识到虚拟机本身在功能

上并不局限于某一种语言。任何能够被转化为字节代码的内容都能在虚拟机上发挥应有的作用。

现在,似乎每个人都在为实现这一目的而着手构建自己的语言。将构建虚拟机方面的工作完全交给 Sun/ 甲骨文公司,而将自己的主要精力放在语法与结构的种种细节上,这已经成为当下极为常见的开发态势。

因此,依附于 JVM 的语言正在不断增多。Ruby 爱好者们乐于谈论 JRuby 在繁重负载下的良好工作状态。Scala 与 Cloture 则可以让开发者们编写出的代码能够在同一套 JVM 中实现比 Java 更多的功能性及更快的运行速度。

甚至连 Java 的拥趸们也开始尝试使用 JVM 却抛开 Java 的做法。比如说 Groovy,这是一款集合了 JVM 及 Java API 的脚本语言。利用 Groovy 进行缩略并同时采用 Grails,大家就能够以类似 Rails 的编码规则进行工作。需要调用 Java 函数库? 闹太套。Groovy 自身的无缝工作设计让我们完全能够在不用大括号的前提下实现 Java 的全部功能性与稳定性。总之,有了它,妈妈再也不用担心我的编程。

编程发展趋势第二位:JavaScript 不再只为 JavaScript

JVM 并不是唯一一款向各类语言敞开大门的跨平台解决方案。JavaScript,现在也不再只局限于 JavaScript 编码的苑囿。比起目前各类繁多的精巧语言交叉编译器,JVM 上的种种 JavaScript 实时引擎更是百花齐放。

编程领域的十一大发展趋势 II

以谷歌 Web 工具包来例。我们大可以像上个世纪九十年代在 Swing 框架上那样编写 Java 代码,只要通过 GWT 编译器的转换,结果就可以顺利运行于拥有 JavaScript 的台式机、智能手机或者平板设备的浏览器中。至于客户端上是否具备 Java 小程序插件或者 JVM 则完全不重要,因为浏览器中的 JavaScript 足以提供必要的支持功能。

CoffeeScript 是这一领域中的后起之秀,这款缩略语言通过帮助讨厌输入各类标点符号的脚本语言用户自动插入必要符号将目标代码转换为 JavaScript 可识别的格式。这个主意可谓深得人心,因此现下已经出现了像 Coco、Parsec-Coffee-Script 以及 Contracts-Coffee-Script 等一系列衍生产品,其中每一款都拥有各具特色的精巧元编程结构,旨在允许用户更便捷地转换代码。

有些衍生产品太过成功,以至于它们自身几乎也成为另一种全新的语言。想象一下,众多 Web 开发者们只需利用 jQuery 敲出可用的代码,而完全不必理会什么 JavaScript 语言的系统知识,这一切是多么美好啊。

如果大家认为上述事例还不足以说明问题,不妨点击这个链接。这里有包括 Ruby、Python、Lisp、Scheme、Haskell 以及 OCaml 等几乎各种语言向 JavaScript 转换的范例。

编程发展趋势第三位: 代码永远不可能独善其身

有位朋友曾经告诉我,他曾在位于密歇根州北部由他父亲与叔叔亲手搭建的小屋中生活过一段时间。这一小片天地只属于他们自己,因为这里的一草一木、一砖一瓦都浸透他们的汗水。在

那里,他们可以轻松地享受夕阳的抚慰并自由自在地沉浸在成就与自豪的情感当中。

编写程序曾经也是这样。按下编译按钮,那么转换过后,代码就会在我们的设备上有序运转。当然,它与操作系统层之间存在交互,但同时它也是实实在在的有形成果,正如朋友所拥有的小屋。快看这个 .EXE 文件,这是我独立完成的作品——在那段美好的时光中,我们完全可以这样骄傲地向朋友们宣布。

然而,这种特征正在消失。我们的代码越来越多地依赖于生态系统。举例来说,许多 PHP 程序员都只是在为 WordPress、Drupal、Joomla 或者其它什么框架创建插件。他们的代码总是同时也不得不以模块化形式出现,以保障与其它模块的并行运作。

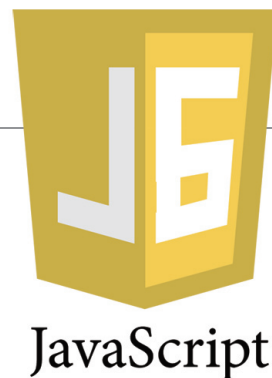
有时具体的单位甚至更小,由代码片段缩小为字符串。例如不少 Drupal 模块就可以通过调整为 PHP 服务。程序员只需要将片段形式的代码填充进去即可,而无需单独创建任何新的内容。

当这一整套机制成为主导时,结果当然可能是积极的。但实际上,真正的结果往往喜忧参半。虽然大家使用的代码片段能够在引擎之下发挥作用,但随之而来的还有大量调试工作。在多数情况下,错误都来自系统的核心部位,那是我们完全意想不到的问题来源。而且关于这部分内容的文档资料也很少,因为基本没人指望程序员能以这样的深度剖析系统故障。

而在最极端的情况下,错误也许来自其他人所引用的代码片段。(未完查看网络原文:

<http://developer.51cto.com/art/201112/305390.htm> ■

JavaScript 6 看上去很美



本文是外刊 IT 评论从 ECMAScript 6 looks promising 这篇文章翻译而来。JavaScript 6 让我很兴奋,这是我看了 David Herman 在 YUIConf 2011 大会上的演讲后的反应。我非常希望这些语言特征

这些新增加的语言特征对于一个使用 JavaScript 的程序员来说,解决了很多以前使用起来很不方便的东西。尽管这个标准中 2013 年才能制定完成(大会上 David 这样说的),谷歌浏览器和火狐浏览器很有可能会提前实现这些特征。

从我来说,下面的这些特征我很感兴趣:

let 关键字

let 关键字能产生一个代码块内的变量范围。因为 var 有一个函数式的范围概念,这有时候容易产生出不注意的错误。以后,我们完全使用 let 来代替 var,从而避免这样的错误。

缺省参数

```
function foo(bar="baz") {  
    console.log(bar);  
}
```

有了缺省参数,省去了很多折腾的代码。

非严格的解构方式

很像 Python 语言,解构时不需要严格匹配。

```
let [x,y] = [3,4,5]; // x=3, y = 4
```

多行模式的字符串表达

你可以简单的使用 ` (反点号) 操作符来声明一个多行字符串。

```
var htmlString = `Say hello to  
multi-line  
strings!`
```

模板化

你可以通过这种方式把 JavaScript 变量嵌入的字符串中:

```
var firstName = "Jack";  
var message = `Hello ${firstName}!`; // "Hello  
Jack!"
```

List 复合操作

同样,这也是一个非常 Python 的结构:

```
let even = [ x for (x in values([1,2,3,4,5,6])) if  
(x %2 === 0) ];
```

使用 values() 能够让 x 来表示 List 里的成员的值,而不是成员变量。你也可以使用新的 for of 语法来实现这样的功能:

```
let even = [ x for(x of [1,2,3,4,5,6]) if (x%2  
=== 0) ];
```

除此之外, map, filter, reduce 等等概念也将会成为新标准的内容之一(有一些功能已经在谷歌浏览器和火狐浏览器里被实现,IE9 也有不少这方面的改进)。

更新:发现这个帖子被顶到了 HN 的首页,我必须要加上遗漏的新标准中的模块系统操作功能。演讲中的例子:

```
import { $ } from "jquery.js"  
  
import { map, each } from "underscore.js"
```

还是一样,全是 Python 的套路!被引用的

什么情况该选择开发Web App?

移动 Web 无所不在

移动 web 是目前唯一的支持各种设备访问的平台,与桌面 web 一样,移动 web 支持各种标准的协议。移动 web 也是唯一一个可供开发者发布移动应用的平台。它将各种移动交互与桌面任务有效地连接了起来。

移动 web 平台具有以下一些优点:容易学习,开发成本低、标准化、使用方便、易于发布。我将这些优点归结为无处不在原则:通常开发简单而且市场广泛的东西往往能够取得成功。

“高质量”也是成功的关键因素之一,而这一点移动 web 这些年还没能做得很好。移动 web 在许多方面还存在一些困难,如解决设备的多样性就是提升移动 web 质量的一个重大挑战。但是,移动 web 遇到的困难远没有 native apps 遭遇的挑战来得复杂。另外,这些挑战将在这几年的时间里得到有效的解决。

众所周知,我是移动 web 的拥护者;然而,我也是第一个承认 native app 重要性的人。开发 native App 可以充分利用设备的特性,而这一点往往是 web 浏览器做不到的,所以对一个产品本身而言,native App 是最佳的选择。

下面几节将讨论一下 native App 的一些主要功能。

什么时候应该开发一个 Native App

◆ 为应用收费

没有任何地方规定开发者不能对一个移动 web app 收取使用费,但是由于某些原因,人们常常认为不能或是不应该对一个 web app 收取费

用。由于历史原因,导致移动设备上付费服务遭遇两大阻力:

◆ 付款方式

在移动设备上输入信用卡号相当麻烦,而且在许多老式设备上也没有安全保障。一种典型的方式是,如果你需要对你的应用收费,你可以与运营商达成协议,让运营商代为你的服务收费。这也意味着,你需要和多个运营商达成合作。这通常是首选的方法,因为许多手机用户可能根本就没有信用卡,比如青少年。

另一种方法是将用户的信用卡信息保存在一个安全的网站上。用户可以通过登录到该网站购买应用服务。这个过程不算特别理想,因为这意味着用户不能直接通过他们的移动设备购买服务了。

◆ 强制分成

移动运营商是会提成的。app 无论是通过运营商还是通过移动设备发布,他们都为应用提供了一套收费机制。这些运营商和移动设备将会提取部分收益,然后将剩余的部分交给应用开发商,这也意味着,开发人员必须遵守他们的市场规则。适应运营商的市场规则通常是非常困难的,需要投入大量的人力资源。相比而言,移动设备的市场规则则简单许多,但是也存在不少的困难。

妨碍运营商和移动设备开发商利益的应用以及服务都将受到阻扰。过去,那些不靠运营商和移动设备开发商运作的网站如果收入过于显眼的話,都逃脱不了被关闭的命运,但是最近,这样的事情鲜少发生了。

什么情况该选择开发 Web App ? II

如果你想为你的 native app 收费,那么你就必须接受这个现实——你必须遵守别人的市场规则,还得放弃部分收益。

◆ 开发游戏

如果你是想开发一个移动游戏(移动游戏是移动市场上最大的一块),那么你需要开发一个 native app。游戏对资源的占用很大,并且需要使用许多设备 API 或平台 API。虽然,现在有几款完全使用 web 技术开发的游戏占有了一定的市场份额,但是和 native app 市场的占有情况相比,还是微不足道的。游戏用户对应用的视觉和操作效果要求很高。移动 web 虽然提供了一些仿真体验,但还远远不能满足用户的需求。

在开发移动游戏时,你需要慎重考虑你的应用需要支持哪些平台。幸运的是,现在有许多工具能够帮助你将你的游戏推向多个平台,但是完成这些工作,还是需要花费大量的人力和物力。

◆ 使用定位功能

下一个功能就是定位功能,可以通过 GPS 或者是信号检测确定用户当前的位置信息。以前只能通过 native app 的 APIs 查看用户的位置信息,但现在大多数主流移动浏览器上都嵌入了 W3C Geolocation API。像 iPhone 或 Android 这样安装了 WebKit 的设备,或是配置了 Opera 或 Mozilla 浏览器的设备,都可以获取用户的位置信息。

我相信定位功能会为 Web 技术带来许多全新的应用。如果能够合理利用 web 浏览器,web 开发商就能使用用户的位置信息和其他内容开发出更加有趣的应用。虽然这在技术上没有太大的困难,但却受到隐私保护条例的限制。我们将

web 浏览器当做是用户进入 World Wide Web 的入口。加入定位功能,意味着在网站中引入了一些敏感信息,这有可能导致严重的后果。但是位置感知应用中显示的位置信息必须经过用户的授权,用户当然有权禁止应用发布自己的位置信息。

◆ 使用摄像头

摄像头可以为你的应用提供丰富的可能性。以往移动 MMS(Multimedia Messaging Service)被用于处理移动照片。换言之,你拍了一张照片后,需要使用 MMS 将它传送给一个服务器,服务器对照片做出相应的处理,并将处理完成的结果通知给你。这个过程是非常耗时的,而且相当复杂,也没有可靠性保障。

通过访问摄像头,native app 开发者能够简化拍照的过程。用户可以直接在客户端对照片做一些简单的处理,只有在有需要的时候才将照片上传给服务器,而且是通过可靠的 HTTP 传输。W3C 正在开发一个访问摄像头的 API,但现在还没有将这部分工作正式整合到浏览器中。

在许多类型的移动 apps 中,摄像头是非常有用的,比如快拍应用、短片拍摄应用等等,摄像头可以用来捕捉许多重要的瞬间。不久的将来,我们可以看到——只要通过摄像头拍摄某个标识,应用程序就能自动完成对标识上的语言转换工作——这个技术在日本已经开始流行起来了。

(未完查看网络原文请点击

<http://developer.51cto.com/art/201112/306812.htm>

注:本文作者为畅销移动 Web 图书作者 Brian Fling,本文为其著作《Mobile Design And Development》的样章节选。 ■

jQuery开发技巧和心得

在这篇文章中,我们将介绍 15 个让你的 jQuery 更加有效的技巧,大部分关于性能提升的,希望大家能够喜欢!

1. 尽量使用最新版本的 jQuery 类库

jQuery 项目中使用了大量的创新。最好的方法来提高性能就是使用最新版本的 jQuery。每一个新的版本都包含了优化的 bug 修复。对我们来说唯一要干的就是修改 tag,何乐而不为呢?

我们也可以使用免费的 CDN 服务,例如, Google 来存放 jQuery 类库。

```
<!-- Include a specific version of jQuery -->
<script src="http://ajax.googleapis.com/ajax/
libs/jquery/1.6.1/jquery.min.js"></script>
<!-- Include the latest version in the 1.6
branch -->
<script src="http://ajax.googleapis.com/ajax/
libs/jquery/1.6/jquery.min.js"></script>
```

2. 使用简单的选择器

直到最近,返回 DOM 元素的方式都是解析选择器字符串,javascript 循环和内置的 javascript API 例如, getElementById(), getElementsByTagName(), getElementsByClassName() 三种方式的整合使用。但是现代浏览器都开始支持 querySelectorAll(),这个方法能够理解 CSS 查询器,而且能带来显著的性能提升。

然而,我们应该避免使用复杂的选择器返回元素。更不用说很多用户使用老版本的浏览器,强迫 jQuery 去处理 DOM 树。这个方式非常慢。

```
$('li[data-selected="true"] a') // Fancy, but
slow
$('li.selected a') // Better
$('#elem') // Best
```

选择 id 是最快速的方式。如果你需要使用 class 名称,那么你最好带上 tag 名称,这样会更快些。特别是在老浏览器和移动设备上。

访问 DOM 是 javascript 应用最慢的方式,因此尽量少使用。使用变量去保存选择器,这样会使用 cache 来保存。性能更好。

```
var buttons = $('#navigation a.button'); //
Some prefer prefixing their jQuery variables with $:
var $buttons = $('#navigation a.button');
```

另外一个值得做的是 jQuery 给了你很多的额外便利选择器,例如, :visible, :hidden, :animated 还有其它,这些不是合法的 CSS3 选择器。结果是你使用这些类库就不能有效地利用 querySelectorAll() 方法。为了弥补这个问题,你需要先选择元素,再过滤,如下:

```
$('.a.button:animated'); // Does not use
querySelectorAll()
$('.a.button').filter(':animated'); // Uses it
```

3. 数组方式使用 jQuery 对象

运行选择器的结果是一个 jQuery 对象。然而, jQuery 类库让你感觉你正在使用一个定义了 index 和长度的数组。

```
// Selecting all the navigation buttons:
var buttons = $('#navigation a.button');
// We can loop though the collection:
for(var i=0;i<buttons.length;i++){
    console.log(buttons[i]); // A DOM
    element, not a jQuery object
}
// We can even slice it:
var firstFour = buttons.slice(0,4);
```

如果性能是你关注的,那么使用简单 for 或者 while 循环来处理,而不是 \$.each(),这样能使你的代码更快。

jQuery 开发技巧和心得 II

检查长度也是一个检查你的 collection 是否含有元素的方式。

```
if(buttons){ // This is always true
  // Do something
}
if(buttons.length){ // True only if buttons
contains elements
  // Do something
}
```

4. 选择器属性

jQuery 提供了一个属性,这个属性显示了用来做链式的选择器。

```
$('#container li:first-child').selector //
#container li:first-child
$('#container li').filter(':first-child').selector
// #container li.filter(:first-child)
```

虽然上面的例子针对同样的元素,选择器则完全不一样。第二个实际上是非法的,你不可以使用它来创建一个对象。只能用来显示 filter 方法是用来缩小 collection。

5. 创建一个空的 jQuery 对象

创建一个新的 jQuery 空间能极大的减小开销。有时候,你可能需要创建一个空的对象,然后使用 add() 方法添加对象。

```
var container = $();
container.add(another_element);
```

这也是 quickEach 方法的基础,你可以使用这种更快的方式而非 each()。

6. 选择一个随机元素

上面我提到过, jQuery 添加它自己的选择器过滤。除了类库,你可以添加自己的过滤器。只需要添加一个新的方法到 \$.expr[':'] 对象。一个非常棒的使用方式是 Waldek Mastykarz 的博客中提到的: 创建一个用来返回随机元素的选择器。

你可以修改下面代码:

```
(function($){
  var random = 0;
  $.expr[':'].random = function(a, i, m, r) {
    if (i == 0) {
      random = Math.floor(Math.random() *
r.length);
    }
    return i == random;
  };
})(jQuery);
// This is how you use it:
$('li:random').addClass('glow');
```

7. 使用 CSS Hooks

CSS hooks API 是提供开发人员得到和设置特定的 CSS 数值的方法。使用它,你可以隐藏浏览器特定的执行并且使用一个统一的界面来存取特定的属性。

```
$.cssHooks['borderRadius'] = {
  get: function(elem, computed, extra){
    // Depending on the browser, read the
value of
    // -moz-border-radius, -webkit-
border-radius or border-radius
  },
  set: function(elem, value){
    // Set the appropriate CSS3 property
  }
};

// Use it without worrying which property the
browser actually understands:
$('#rect').css('borderRadius',5);
```

更好的在于,人们已经创建了一个支持 CSS hooks 类库 (未完请查看网络原文:

<http://developer.51cto.com/art/201111/302522.htm>



点击查看:

<http://developer.51cto.com/art/201102/245390.htm>

高效整洁CSS代码原则

CSS 学起来并不难,但在大型项目中,就变得难以管理,特别是不同的人 CSS 书写风格上稍有不同,团队上就更加难以沟通,为此博主漫凯维奇总结了一些如何实现高效整洁的 CSS 代码原则:



1. 使用 Reset 但并非全局 Reset

不同浏览器元素的默认属性有所不同,使用 Reset 可重置浏览器元素的一些默认属性,以达到浏览器的兼容。但需要注意的是,请不要使用全局 Reset:

```
*{ margin:0; padding:0; }
```

这不仅仅因为它是缓慢和低效率的方法,而且还会导致一些不必要的元素也重置了外边距和内边距。在此建议参考 YUI Reset 和 Eric Meyer 的做法。我跟 Eric Meyer 的观点相同,Reset 并不是一成不变的,具体还需要根据项目的不同需求做适当的修改,以达到浏览器的兼容和操作上的便利性。我使用的 Reset 如下:

```
/** 清除内外边距 **/  
body, h1, h2, h3, h4, h5, h6, hr, p,  
blockquote, /* structural elements 结构元素 */  
*,  
dl, dt, dd, ul, ol, li, /* list elements 列表元素 */  
*,  
pre, /* text formatting elements 文本格式元素 */  
*,  
form, fieldset, legend, button, input,  
textarea, /* form elements 表单元素 */  
th, td, /* table elements 表格元素 */  
img, /* img elements 图片元素 */ {  
    border: medium none;  
    margin: 0;  
    padding: 0;  
}  
/** 设置默认字体 **/
```

```
body, button, input, select, textarea {  
    font: 12px/1.5 '宋体', 'Tahoma', 'Arial',  
    helvetica, sans-serif;  
}  
h1, h2, h3, h4, h5, h6 { font-size: 100%; }  
em { font-style: normal; }  
/** 重置列表元素 **/  
ul, ol { list-style: none; }  
/** 重置超链接元素 **/  
a { text-decoration: none; color: #333; }  
a: hover { text-decoration: underline;  
color: #F40; }  
/** 重置图片元素 **/  
img { border: 0px; }  
/** 重置表格元素 **/  
table { border-collapse: collapse; border-spacing: 0; }
```

2. 良好的命名习惯

无疑乱七八糟或者无语义命名的代码,谁看了都会抓狂。就像这样的代码:

```
.aaabb{margin:2px;color:red;}
```

我想即使是初学者,也不至于会在实际项目中如此命名一个 class,但有没有想过这样的代码同样是有问题的:

```
<h1>My name is <span class="red  
blod">Wiki</span></h1>
```

问题在于如果你需要把所有原本红色的字体改成蓝色,那修改后就样式就会变成:

```
.red{color:bule;}
```

这样的命名就会很让人费解

高效整洁 CSS 代码原则 II

同样的命名为 .leftBar 的侧边栏如果需要修改成右侧边栏也会很麻烦。所以,请不要使用元素的特性(颜色,位置,大小等)来命名一个 class 或 id,您可以选择意义的命名如: #navigation{...}, .sidebar{...}, .postwrap{...}

这样,无论你是否修改定义这些 class 或 id 的样式,都不影响它跟 HTML 元素间的联系。

另外还有一种情况,一些固定的样式,定义后就不会修改的了,那你命名时就不用担忧刚刚说的那种情况,如

```
.alignleft{float:left;margin-right:20px;}
.alignright{float:right;text-align:right;margin-left:20px;}
.clear{clear:both;text-indent:-9999px;}
```

那么对于这样一个段落

<p class="alignleft"> 我是一个段落! </p>

如果需要把这个段落由原先的左对齐修改为右对齐,那么只需要修改它的 className 就为 alignright 就可以了。

3. 代码缩写

CSS 代码缩写可以提高你写代码的速度,精简你的代码量。在 CSS 里面有不少可以缩写的属性,包括 margin, padding, border, font, background 和颜色值等,如果您学会了代码缩写,原本这样的代码:

```
li{
  font-family:Arial, Helvetica, sans-serif;
  font-size: 1.2em;
  line-height: 1.4em;
  padding-top:5px;
  padding-bottom:10px;
  padding-left:5px;
}
```

就可以缩写为:

```
li{
  font: 1.2em/1.4em Arial, Helvetica, sans-serif;
  padding:5px 0 10px 5px;
}
```

如果您想更了解这些属性要怎么缩写,可以参考《常用 CSS 缩写语法总结》或者下载 CSS-Shorthand-Cheat-Sheet.pdf。

4. 利用 CSS 继承

如果页面中父元素的多个子元素使用相同的样式,那最好把他们相同的样式定义在其父元素上,让它们继承这些 CSS 样式。这样你可以很好的维护你的代码,并且还可以减少代码量。那么本来这样的代码:

```
#container li{ font-family:Georgia, serif; }
#container p{ font-family:Georgia, serif; }
#container h1{font-family:Georgia, serif; }
```

就可以简写成:

```
#container{ font-family:Georgia, serif; }
```

5. 使用多重选择器

你可以合并多个 CSS 选择器为一个,如果他们有着共同的样式的话。这样做不但代码简洁且可为你节省时间和空间。如:

```
h1{ font-family:Arial, Helvetica, sans-serif;
font-weight:normal; }
h2{ font-family:Arial, Helvetica, sans-serif;
font-weight:normal; }
h3{ font-family:Arial, Helvetica, sans-serif;
font-weight:normal; }
```

可以合并为

```
h1, h2, h3{ font-family:Arial, Helvetica,
sans-serif; font-weight:normal; }
```

(未完查看网络原文:

<http://developer.51cto.com/art/201112/305667.htm> ■

■ 编者按

最新的 Android 4.0 有很多新特性,51CTO 特约了著名手机厂商 Android 开发工程师、最牛网站长作为本站专栏作者,为各位网友撰写 Android 4.0 开发相关文章。本文我们将探讨 Android 4.0 的图形硬件加速及绘制技巧如何得到了更好的提升。

Android 4.0图形硬件加速及绘制技巧

从 Android 3.0 开始,Android 2D 的绘制流程就设计为能够更好地支持硬件加速。使用 GPU 的 View 在 Canvas 上进行画的操作时都会使用硬件加速。在最新的 Android 版本里,图形硬件加速及绘制技巧得到了更好的提升,51CTO 特约了最牛网站长作为本站专栏作者,为各位网友撰写 Android 4.0 开发相关文章。

常表现在不可见的元素异常和错误的像素渲染,为了解决这个问题 Android 可以让你选择启动或者禁用以下级别的硬件加速:Application Activity Window 和 View 。

1.1.1 Application 级别

在你的 Android Manifest 文件中添加 属性标记,以便为整个应用程序使用硬件加速。



1.Android 4.0 硬件加速的使用

1.1 硬件加速控制级别 Android Fragment

启用硬件加速的最简单方法就是为整个系统打开硬件加速的全局设置。如果你的程序是标准 View 或者是 Drawable 则硬件加速的全局设这并不会造成不良的影响。然而硬件加速并不支持所有 2D 画的操作,所以开启硬件加速可能会对使用自定义组件的应用程序造成影响,问题常

1.1.2 Activity 级别

如果你的应用程序不能在 Application 应用级别表现良好的话,则可以使用对 Activity 进行单独控制。要启动或者禁用一个 Activity 的硬件加速,你可以使用 activity 的 android:hardwareAccelerated 属性。下面的一个列子使整个 Application 启用硬件加速,但是对一个 Activity 禁止使用硬件加速。

Android 4.0 图形硬件加速及绘制技巧 II

1.1.3 Window 级别

如果你需要更细粒度的控制,你可以通过如下代码给 window 进行加速。Android Loader

注意:现阶段你不能在 Window 级别对它禁用硬件加速。

1.1.4 View 级别

我们可以对单独的 View 在运行时阶段禁用硬件加速。

注意:现阶段不能够在 View 级别进行硬件加速。

1.2 判断一个 View 是否已经启用了硬件加速

有时候我们需要知道一个应用程序是否已经启用了硬件加速,特别是针对一些自定义控件。因为你的应用程序做了很多自定义“画”的操作,但并不是所有的过程都支持新的“画”的渲染过程。

有两种不同的方法来检查 Application 是否启用了硬件加速:

1.2.1 使用 View.isHardwareAccelerated() 如果返回 true 则可以说明这个 View 所在的窗口已经启用了硬件加速。

1.2.2 Canvas.isHardwareAccelerated() 如果返回 true 则说明这个 Canvas 已经启用了硬件加速。

如果你必须要在你的绘画代码中进行是否已经加速的检查,如果可能的话请使用 Canvas.isHardwareAccelerated() 来代替 View.isHardwareAccelerated()。当一个 View 是存在于一个已经加速的 Windows 上时,任然可以使用没有硬件加速的 Canvas 进行绘画,这场发生在,比

如,当我们把一个 View 画到 Bitmap 上然后用作缓存。

2.Android 4.0 的绘制模型

当开启了硬件加速,Android 框架将会使用一种新的绘制模型,这种模型将会使用显示列表把你的应用显示到屏幕上。要完全理解显示列表和他们如何影响你的应用程序,理解 Android 4.0 如何在非硬件加速的情况下如何绘制 Views 是很有必要的,下面将分别介绍软件加速和硬件加速。

2.1 基于软件的绘制模型

在基于软件绘制模型中,View 的绘制遵循以下两步:1. 使整个控件层级无效。2. 对层级进行绘制。

当一个应用程序需要更新它 UI 的一部分时,它将会调用内容发生改变的 View 的 invalidate() 方法(或者 invalidate 的变体)。Invalidate 的消息按照 View 的层级关系向上传递用以计算需要重画的部分(即脏区域)。然后 Android 系统会对和脏区域有交集的所有 View 进行绘制,不幸的是这种模型中有两个缺点:

2.1.1 在这种模型中当在不同的层进行画的时候,会额外执行很多代码。

例如一个 Button 是位于另外一个 View 之上,当对 Button 调用 Invalidate() 时,Android 就会对这个 View 进行重绘,即便这个 View 没有发生任何变化。

2.1.2 第二个问题是这种绘制模型会隐藏你 Application 中的 Bug。

因为 Android 系统会对和脏区域有交集的 View 进行重绘,在这种情况下如果一个 view 的

Android 4.0 图形硬件加速及绘制技巧 III

内容发生了改变,即便这个 View 的 `Invalidate()` 的方法并没有得到调用,它也可能被重绘。你便会依赖调用了 `invalidate()` 的其他的控件以便获得正确的行为,因此每当你的 Application 发生改变时,这种行为多要随之发生改变。也是基于此因,在你的自定义控件中你必须不断地调用 `invalidate()` 方法,当你的数据或者是状态会影响 View 的绘制代码时。

注意: Android 的 View 当它们的属性发生改变时会自动的调用 `Invalidate()`。比如,你改变一个 Textview 的背景或者是它的文本。

2.2 基于硬件加速模型

Android 系统仍然通过 `invalidate()` 和 `draw()` 去请求屏幕更新和重新渲染,但是实际处理画的方式是不同的。不是立即执行画的命令,Android 而会将所有画的命令记录在一个显示列表里面,这个显示列表包含了输出的 View 层级的绘制代码。还有一个优化就是 Android 在显示列表中只会记录和更新显示层级中通过调用 `invalidate()` 函数被标记为“脏”的 view。没有被请求刷新的 view 可以通过重新请求先前的显示列表以便重画。新的绘制模型包括有三个步骤: 1. 禁用整个 View 层级。2. 记录和更新显示列表。3. 绘制显示列表。

使用这个模型你不能依赖一个 View 和脏区域有交集就会执行 `draw()` 方法。要确保 Android 系统记录了一个 View 的显示列表,你必须调用 `invalidate()` 方法,如果忘记了调用刷新,会使 View 即便是发生了改变后也会看起来相同,这是一个

比较容易发现 bug 的方式。

使用显示列表的方式对动画的表现也是很有好处的,因为设置指定的属性值,比如透明度或者旋转,就不需要请求刷新目标 View(这将自动执行)。这项优化也应用于有显示列表的 Views(启用了硬件加速的 View),例如,现在有一个 LinearLayout 包含了一个 ListView 和 Button, listview 在 button 的上面。这时候 LinearLayout 的显示列表如下所示:

- ◆ `DrawDisplayList(ListView)` ;
- ◆ `DrawDisplayList(Button)` ;

假设你现在你想更新这个 Listview 的不透明度,在设置 Listview 的 `setAlpha(0.5f)` 属性之后, LinearLayout 的显示列表应该包含如下:

- ◆ `SaveLayerAlpha(0.5)`
- ◆ `DrawDisplayList(ListView)`
- ◆ `Restore`
- ◆ `DrawDisplayList(Button)`

这时候绘制 Listview 的复杂过程就会省略了,取而代之的是简单的更新了 LinearLayout 的显示列表。如果一个应用程序并没有启用硬件加速, Listview 和它的父 view 的画的代码都会重新执行。

3.Android 4.0 View 的层

3.1 层的分类

所有的 Android 版本都有能力对离屏缓冲进行渲染,或者是使用 View 的绘制缓冲,或者是使用 `Canvas.saveLayer()` 函数。离屏缓冲或者 Layer 能够有很多种应用,例如能使处理复杂 view 的动

Android 4.0 图形硬件加速及绘制技巧 IV

画效果或者应用一些合成效果都有更好地表现。例如你可以通过 `Canvas.saveLayer()` 的方式来对 `View` 做一个渐入渐出效果同时把它渲染到 `Layer` 中,然后再加上不透明效果合成后显示到屏幕上。

由 Android 3.0 开始你就能够通过 `View.setLayerType()` 方法对何时以及如何使用层有了更多的控制,这个 API 具有两个参数一个是你想使用的层类型,另外一个可选参数 `Paint` 表明了 `Layer` 是如何被叠加的。你可以把 `Paint` 参数应用到颜色过滤上,特别是混合模式或者是对一个 `layer` 进行不透明效果。一个 `View` 可以使用如下的三种 `layer` 类型之一:

◆ `LAYER_TYPE_NONE`: 这个 `View` 将被按普通的方式进行渲染,但是不会返回一个离屏的缓冲,这个是默认的行为。

◆ `LAYER_TYPE_HARDWARE`: 如果这个应用被硬件加速的话,这个 `View` 将会在硬件中渲染为硬件纹理,如果应用程序并没有被硬件加速,则其效果和 `LAYER_TYPE_SOFTWARE` 是相同的。

◆ `LAYER_TYPE_SOFTWARE`: 此 `View` 通过软件渲染为一个 `bitmap`。

3.2 层的使用

使用层的类型取决于你的目的:

3.2.1 性能

使用硬件层来渲染一个 `View` 成为硬件纹理。一旦一个 `View` 被渲染为一个层,它的绘制代码将不会得到执行,直到你调用了 `invalidate()` 函数。

对于一些动画,比如透明动画可以直接应用到一个层上,这是 GPU 最有效率的使用方式。

3.2.2 显示效果

使用硬件或者软件层和 `Paint` 来对一个 `View` 进行特殊的视觉处理,例如你可以对一个 `View` 通过使用 `ColorMatrixColorFilter` 来实现黑白效果。

3.2.3 兼容性

使用软件层类型会强制使一个 `view` 在软件中被渲染。如果一个 `view` 是硬件加速的话(比如你设置整个应用程序是硬件加速的话),同时有渲染的问题,这是一种很简单的方式来限制硬件绘制流程。

3.3 View 的层和动画的关系

当你的应用程序已经使用了硬件加速的话,硬件层能够带来更为快速和更为平滑的动画效果。当对一个复杂的 `View` 进行动画操作时,因为要进行很多的画操作,所以并不可能总是能达到 60 帧每秒。在这种情况下可以通过硬件层来渲染为硬件纹理来提高性能。硬件纹理操作可以用作对一个 `view` 进行动画操作,当进行动画的时候可以减少对 `View` 自身频繁的重绘。除非你改变这个 `view` 的属性(调用 `invalidate()` 方法)或者你手动的调用 `invalidate()`。如果在你的应用中运行一个动画,但是并没有得到你想要的平滑效果,可以考虑为你要动画的 `view` 开启硬件层。

当一个 `View` 通过硬件层返回时,当所有的层叠加后最终的画面显示在屏幕时, `View` 一些属性会被同时被处理。设置这些属性是十分有效率

Android 4.0 图形硬件加速及绘制技巧 V

如下的属性将影响层的叠加,设置这些属性将会使 View 自动请求刷新,而且不需要对 View 进行重绘。

- ◆ alpha: 改变层的透明度。
- ◆ x, y, translationX, translationY: 改变层的位置
- ◆ scaleX, scaleY: 改变层的大小
- ◆ rotation, rotationX, rotationY: 在 3D 空间内改变层的方向
- ◆ pivotX, pivotY: 指定它进行变形的原点位置

这些属性是通过 ObjectAnimator 对象对一个 view 进行动画操作时所使用的,如果你想访问这些属性,直接调用这些属性的 setter 或者 getter 方法,例如想改变 View 的 alpha 则直接调用 setAlpha()。

如下的代码片段显示了一个 View 通过 Y 轴进行 3D 旋转。

因为硬件层会消耗视频的内存,强烈的推荐你在作动画的时候启用他们,当动画完成了之后禁用他们,你可以通过动画监听来完成这些。

4 Android 4.0 提示和技巧

切换到硬件加速 2D 图形可以立即增强表现,但是你还是需要通过如下的建议来设计你的应用程序来更有效率的使用 GPU。

4.1 减少你程序中使用 View 的数量

你系统中画的 view 的数量越多,你的程序就会越慢,在软件绘制的流程也是一样的,减少 view 的数量是优化你 UI 的一个最简单的方法。

4.2 避免过多绘制

不要过多的叠加层,当一个 View 被其他 view 完全遮挡住了的话,最好把被遮挡的 view 移除掉。如果你需要绘制不同的层做一个叠加效果的话,考虑把这些层合并为一个层。

就现在的硬件来看,有一个好的经验就是动画的每帧不要绘制多余屏幕像素 2.5 倍的像素数量 (bimap 中的透明像素也计算在内)。

4.3 不要在绘制的方法中创建绘制对象

一个常见的错误就是当绘制方法被调用的时候,每次都要创建一个新的 Paint 或者 Path。这将迫使垃圾回收器过于频繁的运行,这将对缓冲和硬件的绘制造成影响。

4.4 不要过于频繁的修改形状

以复杂的 shapes, path 和旋转为例,这些绘制都会用到纹理的遮罩。每当你创建或者修改一个 path, 硬件渲染过程都会创建一个新的遮罩,这耗费的代价是相当大的。

4.5 不要过于频繁的修改 bitmap

每当修改一次 bitmap 的内容,当你下次再绘制它的时候都会以 GPU 的纹理形式上传一次。

4.6 要小心使用 alpha 通道

当你使用 setAlpha, AlphaAnimation 或者 ObjectAnimator 设置一个 View 的透明效果时。它将需要 2 倍离屏的渲染缓冲填充率,当应用一个 alpha 到一个大的 View 上的时候,考虑设置 view 层的类型为 LAYER_TYPE_HARDWARE。

51CTO 原文地址

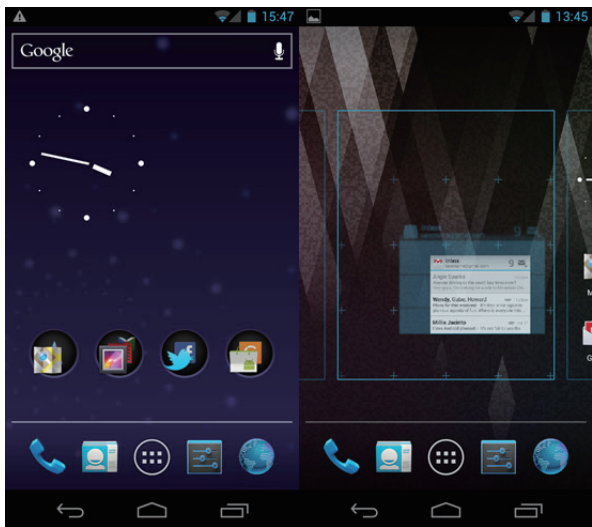
<http://mobile.51cto.com/hot-302203.htm>

与iOS的较量 Android 4.0功能评测

Android 移动设备操作系统中从来没有哪一款像其 4.0 版本或者被称为冰淇淋三明治的这套系统受到如此程度的热切期待,而究其原因,鲜明的特色想必是最主要的卖点。其实这种状态也不难解释:Android 系统虽然在诸多方面已经将 iOS 远远抛在身后,但它的用户界面在美观与格调上的薄弱始终令人略感遗憾。

冰淇淋三明治(简称 ICS)究竟味道如何?

Android 移动设备操作系统中从来没有哪一款像其 4.0 版本或者被称为冰淇淋三明治的这套系统受到如此程度的热切期待,而究其原因,鲜明的特色想必是最主要的卖点。其实这种状态也不难解释:Android 系统虽然在诸多方面已经将 iOS 远远抛在身后,但它的用户界面在美观与格调上的薄弱始终令人略感遗憾。



主屏幕以及蜂窝风格的黑色边框会在图标被拖动时显示出来

ICS 同时也将 Android 系统的风格与使用体验统一到手机与平板设备上,而之前的此类设备,尤其是三星 Galaxy 平板在风格统一性上则较为逊色。其实我们不妨将 ICS 视为蜂窝的 3.0 版本,谷歌对其细心打磨,而没有急于让其与苹

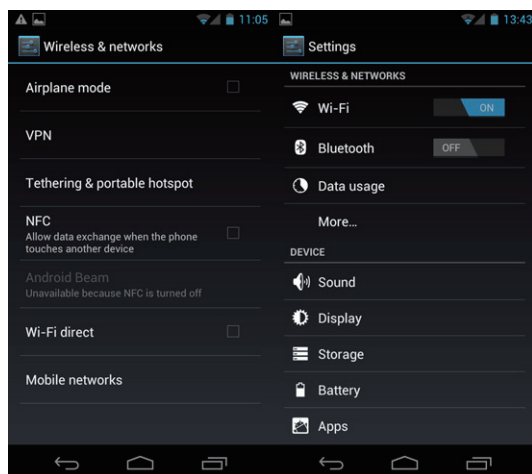
果的 iPad 一争高下。让我们从外观开始,看看 Android4.0 带来的新风貌。只要稍微关注,大家就不难发现 ICS 在保持了其一贯的蜂窝风格主题的基础上,在设计方面较之 Android 的其它早期版本有了一大飞跃。结合从 iOS 与 WinPho7 上借鉴到的灵感,新的用户界面现在满载充满视觉冲击力的动画效果,似乎令人感到在自己的触摸操作下,整个界面都鲜活灵动。尤其是在添加或删除主屏幕上的项目时,这种全新外观风格体现得更加淋漓尽致。



窗体内容与尺寸调节部件

在整个操作过程中,始终都会有蓝色边框与图标区域标记提示大家挪动的预期效果,这与蜂窝系统完全相同。遗憾的是我们无法再像原先那样通过长按屏幕调用该功能,现在只能从应用程序托盘中启动。程序托盘如今也以旋转于窗体调节目录的滚动菜单中。

与 iOS 的较量 Android 4.0 功能评测 II

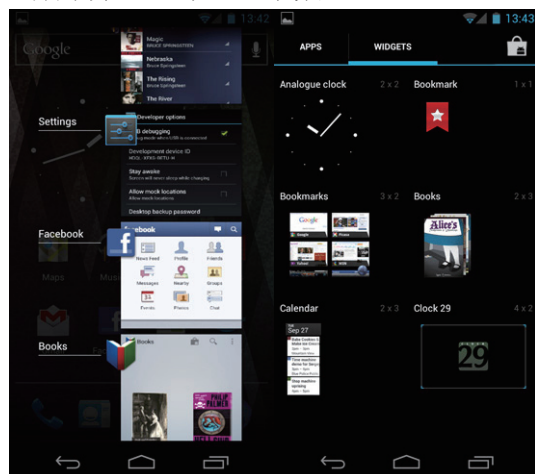


网络与设置

从审美角度来讲,这种做法可以令屏幕内容看起来更清爽,进而令人感觉手机屏幕占整体机身的比例也有所提高。但这又不禁令我好奇,ICS 在那些尚未取消功能键的老机型上会有怎样的显示效果。

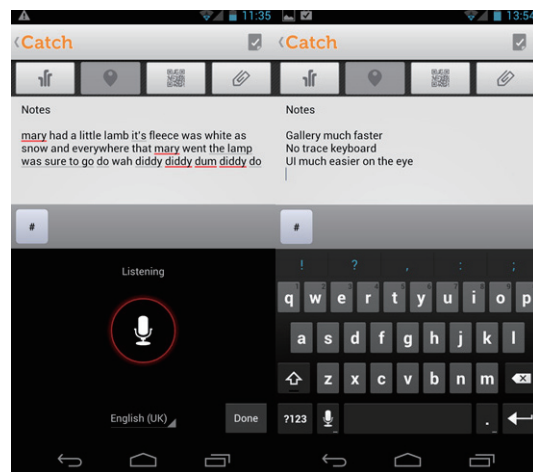
后台信息

现在菜单按钮会根据当前文本与应用程序的不同改变自身在屏幕上的显示位置。有时它出现在屏幕顶端应用程序菜单处,有时在底端,有时则位于毗邻程序运行窗口的系统按钮处。好吧,大家不妨称我为保守派,但我仍然认为菜单按钮始终保持在同一位置更为贴心。



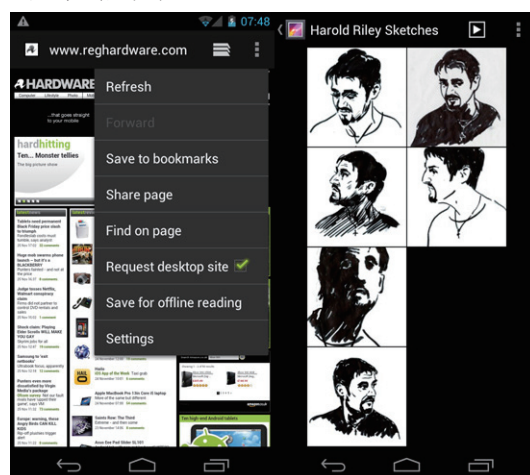
部件选择与最新应用程序列表

这一新排布最大的好处在于让用户得以直观地检视当下有哪些应用程序正处于后台运行状态,并可以将图标拖出屏幕以关闭对应程序。拖出操作的具体选项归入通知栏,大家可以在那里单独取消关闭确认选项。



语音与输入

ICS 系统的虚拟键盘在外观上与 Android v2.3 姜饼系统并无太大区别,但实际上它使用起来更便捷,而且现在也具备了 iOS 键盘那得天独厚且不可思议的精确性。另外,只需键入一小段开头,系统将自动弹出过去曾经输入过的相似内容。如果比起手动输入,大家更乐于尝试语音功能,那么无疑在 Siri 的带动下,语音识别系统已经得到了大幅改善。



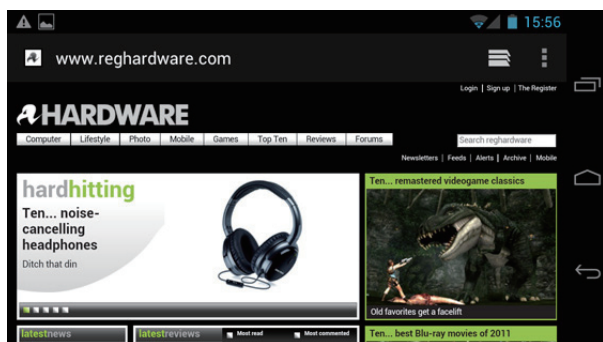
与 iOS 的较量 Android 4.0 功能评测 III

更灵敏的图像反应以及浏览器调整

新的图片显示机制不仅显示效果更好,处理速度也更快。不只是更快,应该说是快得多。假如我们的数据连接处于正常状态,那么 Picasa 图片库将会以调用本地文件的速度显示出图像。该系统的强大性能使得添加新媒体文件的处理速度也得到了显著加强,我们不再需要为偶尔出现的文件调用等待而坐着发呆了。

可视化电话簿

Android 网页浏览器同样得到了良好的改善。它不仅获得了确切的额外负载提速,而且经过长久的等待,我们终于获得了对应选项,使得网页可以在默认状态下以桌面形式显示。页面现在也能够直接在浏览器菜单中加以保存,以待脱机时读取。



浏览器仍不具备 Flash 播放器

有了相机应用程序的导航功能,我们似乎从姜饼系统的石器时代一跃跨入二十一世纪新纪元。新的界面改善卓著,而且其中内置的全景模式效果也出奇理想。在拍摄视频时点击屏幕,即可完成如相机一样的拍照操作,这项功能我认为在未来很可能会派上大用场。在 Gmail 以及 Google+ 之外,ICS 系统的电子邮件及社交网络前端仍旧相当薄弱。尽管电子邮件客户端比起以

往略有起色,但每次当我试图打开合并后的总体收件箱时,应用程序都会崩溃。此外,我的三星 Galaxy Nexus 手机无法添加 Facebook 账户,因此我还不清楚这款名为 People 的全新电话簿程序到底在集成管理各类账户方面有哪些优点。



拨号面板有点单调 数据使用情况非常直观

全面地对新功能进行总结, Android playbook 中倒是有几项有趣的新内容。首先要提到的就是 Face Unlock 利用面部识别技术对手机进行解锁。令人印象深刻的是,该系统在识别我的身份及相关权限方面毫不含糊,成功率高达百分之百;其他人或者我本人的大幅照片也没能骗得过它。

综述

在 ICS 的推动下, Android 系统迎来了其史无前例的巅峰时刻。在外观与使用感受方面,这款系统俨然已经可以与 iOS 分庭抗礼。不过新的外观也带来新的问题,即我们是否还需要来自第三方的视觉主题? 我个人认为, Android 4.0 已经让这一切修饰工具变成可有可无的废物。网页浏览器、键盘、相机管理程序以及媒体播放器,这一切都得到了彻底改善,这种改善不仅指外观,连处理速度与使用便捷性方面也有很大提高。■

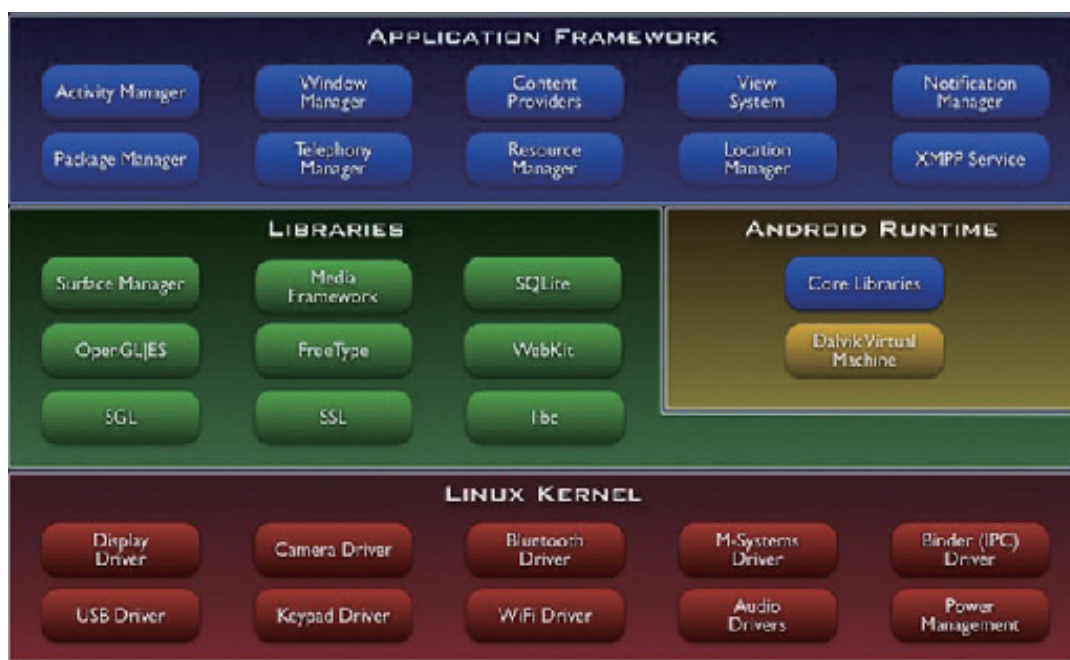
开发者必知：提升Android应用开发性能的十大要点

随着任何一个移动平台不断发展、围绕它的应用程序不断改进,要想开发成功,质量变得至关重要。如今,用户们要求他们决定安装到自己设备上的 Android 应用程序反应迅即、性能合理。如果那些应用程序未能提供令人兴奋的功能和稳定的用户体验,那么很快就会被卸载掉。本文将介绍 Android 提升开发性能十大要点。

虽然 Android 智能手机和平板电脑的速度一天比一天快,但是开发者必须记住一点:他们开发的应用程序仍在一种资源紧张的环境下运行,这种环境主要依赖性能与最新的台式机或笔记本电脑无法比拟的电池和处理器。下面介绍的一些方法可以帮助你的应用程序“瘦身”或者说“Android 提升开发性能十大要点”,以便它们在今天和明天的 Android 设备上最顺畅地运行。

资源使用方面,如果你打开了资源,要记得关闭资源。要尽量晚地获取,尽量早地释放。这些由来已久的编程准则同样适用于你的 Android 应用程序,如果它们使用底层的设备服务,更是如此。

比如说,假设你编写的一个应用程序依赖于位置的服务。除非你绝对有必要,否则不要开始注册、获取位置最新信息;而且要确保,一旦你不再需要这些信息,就要取消获取最新信息的注



编程五大要点

要点一：首先要有良好的编程习惯

要成为一名优秀的资源管理员;既要运用常识,还要使用公认的算法和标准的设计模式。在

册。这将帮助你避免不必要地耗费设备电池电量或占用系统资源。

要点二：让阻塞操作远离主用户界面线程
想确保你的应用程序运行起来很灵活,就要

开发者必知：提升 Android 应用开发性能的十大要点 II

使用 AsyncTask、线程、IntentService 或自定义后台服务来处理脏活。应使用装入器来简化装入时间长的数据（如游标）的状态管理。你无法容忍你的应用程序在某个操作正在处理的时候出现滞后或停顿。

如果某个操作很费时间和资源，就要卸载这部分操作、对它进行异步处理，那样你的应用程序仍保持响应迅即，用户可以处理他们的事务。这个原则适用于下列操作：磁盘读写，访问内容提供方、数据库和互联网，以及解析和其他长时间的任务。

要点三：使用最新的 Android 软件开发工具包 (SDK) 版本、应用编程接口 (API) 和最佳实践

确保你开发的应用程序是最新的，因而要使用 Android 平台提供的最新工具。随着 Android 平台不断发展，它也在不断改进。一些功能可能已被弃用，或者换成了更好的功能。核心 API 得到了修正版 (bug fix) 和性能改进。已经引入了装入器等新的 API，帮助开发者编写出运行更稳定、响应更迅即的应用程序。

你知道可以启用 Android 3.0 应用程序中的硬件加速功能吗？赶紧启用吧！要明白最佳实践会随着时间的变化而变化。明智的开发者密切关注 Android 平台的新功能、哪些功能不再被推荐。

要点四：考虑使用限制模式 (Strict Mode)

你可以使用名为限制模式 (StrictMode) 的 Android API，帮助你查明哪里违反了几个良好的编程习惯。StrictMode 会帮助你确认你的应用程序是不是存在内存泄漏，并且检测你的应用程序

是不是在试图执行长时间的阻塞操作，这些操作应该被卸载到线程或别的渠道。Android 2.3 里面引入 StrictMode 类 (android.os.StrictMode)。

要点五：在发布应用程序之前，禁用或尽量少用调试和诊断

如果你的 Android 应用程序开发起来需要一些时间，你可能已将一些日志和调试代码嵌入到了应用程序中。写入到日志及其他此类输出系统给性能带来了影响。确保在发布应用程序之前，尽量少用或完全禁用这些功能。过来的开发者数量才会慢慢上升。

现在不妨说说如何运用良好的用户界面设计原则，让你应用程序的屏幕更快速地装入：

UI 设计五要点

要点六：确保你设计的布局简单、简练和浅层

简单的屏幕有助于阅读起来最轻松，而简单的布局装入起来最快速。你不应该过于深层地嵌套你的布局，或者用不必要的过多视图 (View) 控件塞满屏幕。花些时间来开发用户可以高效使用的简练用户界面，而不是试图把太多功能塞入到单单一个屏幕上。这不但有助于提升应用程序的性能，还有助于让你的应用程序对用户来说更高效。

Fragments 有助于在不影响灵活地针对不同类型的设备进行设计的情况下，划分用户界面功能。

要点七：让你应用程序的资源适合目标设备

添加适合特定设备配置的资源，那样它们就

开发者必知：提升 Android 应用开发性能的十大要点 III

能尽可能高效地装入。我们在谈论图形资源时,这点尤为重要。如果你添加了可利用的庞大图像资源,需要装入和调整大小,就无法有效地使用其他的应用程序资源。

另一个要点就是,如果你准备你的应用程序可以在许多设备上运行,为了让应用程序软件包文件保持合理的大小,应该最初只添加运行应用程序所需要的核心资源,然后让应用程序下载适合该设备的内容。

要点八：使用 Hierarchy Viewer 工具

Hierarchy Viewer 工具可以帮助你调试你的应用程序布局。它还提供了宝贵的分析信息,以便了解布局里面的每一个视图控件测量、渲染和绘制要花多少时间。只有准确找到了问题的根源,问题解决起来才容易。

要点九：使用 layoutopt 工具

Layoutopt 工具是一款简单的命令行工具,它可以帮助你找到不必要的控件嵌套以及缩减布局资源的其他方法,以便尽量减少资源的使用。它让你可以了解哪些布局控件可能是多余的或不必要的。控件越少、布局层次越浅,性能就越好。

最后,你认为你的应用程序做到了最好吗?现在该对它测试一下了。

要点十：使用 Traceview 及其他 Android 工具进行分析

Android SDK 随带了许多工具,可用来对你的应用程序进行分析。其中最流行的工具恐怕莫过于 Traceview,这款图形化工具可以帮助你调试和找到应用程序中的性能瓶颈。

结束语

有许多方法可以帮助你提升 Android 应用程序的性能。有些需要使用特定的算法,有些依赖切实可行的调试和性能监测技巧。幸好,Android 平台随带众多免费的实用工具,可以帮助你查明和消除应用程序里面的性能问题。既然有了这些工具,就要立马使用! ■

51CTO 原文地址

<http://mobile.51cto.com/hot-305379.htm>

■ 开心一刻

爱好：

iPhone 用户三大爱好：游戏、拍照、发微博。Android 用户三大爱好：刷机、重启、换电池。Symbian 用户三大爱好：摔手机、摔手机、摔手机。山寨用户三大爱好：凤凰传奇、凤凰传奇、凤凰传奇。

砸核桃

在地铁里,看见周围有位美女掏出了 iPhone, 然后一个文艺青年也跟掏出了 iPhone, 一商务男默默的看了一眼,掏出了 iPad, 这个时候,我鬼魅狂狷的一笑,掏出了我的诺基亚,在过道里砸开了一个核桃,于是整个场面被我 hold 住了。

拒绝 iPhone 4S 的六大原因

1. 不如 iPhone 5; 2. 不如 iPhone 6; 3. 不如 iPhone 7; 4. 不如 iPhone 8; 5. 不如 iPhone 9; 6. 不如 iPhone 10。

盗墓

Windows Phone 7 团队在后院埋葬了 iPhone4 和 Blackberry9870。下午当地警局就收到来自微软报案有人盗墓。

■ 编者按

Lua(葡萄牙语中意为“月亮”)首次发布于 1993 年,旨在成为一门嵌入式的解释性脚本语言。它具有小巧、快速、非常简洁的优点,还异常灵活,因而得以进入一系列广泛的商业应用领域。

《Network World》主编谈“Lua 编程语言”

作为著名 IT 杂志的《Network World》主编马克·吉布斯(Mark Gibbs)有一段时间没有谈论编程语言了,这次马克将介绍免费、开源的 Lua 语言,这门语言采用了麻省理工学院(MIT)许可证。

以下为全部译文:

Lua(葡萄牙语中意为“月亮”)首次发布于 1993 年,旨在成为一门嵌入式的解释性脚本语言。它具有小巧、快速、非常简洁的优点,还异常灵活,因而得以进入一系列广泛的商业应用领域。

比如说,Adobe 的 Photoshop Lightroom 其用户界面就是用 Lua 编写的;罗技的 Squeezebox 音乐播放器和 Damn Small Linux(DSL)操作系统也是用 Lua 编写的。据维基百科解释,DSL 使用“Lua,为命令行实用程序提供了对桌面友好的界面,又不占用大量的磁盘空间。”

我得扯开话题,特别指出这一点:Damn Small Linux 眼下步履蹒跚,这个观点出自维基百科关于 DSL 的那篇文章,还有许多人在博文上的留言。维基百科称:“由于该项目的发起者与主要开发者之间发生内讧,DSL 的开发似乎陷入了停滞不前的困境;这个项目前途未卜,这让用户们大为失望。”的确让人失望;而鉴于 DSL 网站似乎死气沉沉(这款操作系统的 4.4 版本发布后,网站在 2008 年似乎没有了人气),祝 DSL 走好。

不管怎样,Lua 学起来很容易,你可以从 Lua

网站下载它,它有多个实现版本。

用 Lua 编写的第一个经典程序相当普通:

```
print("hello")
```

这个命令输出的当然是“hello”。更有意思一点的 Lua 是个阶乘计算器:

```
function fact (n)
  if n == 0 then
    return 1
  else
    return n * fact(n-1)
  end
end

print("enter a number:")
a = io.read("*number") -- read a number
print(fact(a))
```

但也是更有意思一点而已。

这段代码来自 Roberto Ierusalimschy 所著的《Lua 编程》,该书是针对 Lua 5.0 编写的。现在,Lua 的最新版本是版本 5.1。这本书有点过时了,但仍不失为一本实用的入门书。

Lua 是一门引人入胜的语言,有许多示例,可以访问 Luaforge 网站。

你可以试一下交互式在线版。如果你需要更多最新的说明文档,市面上有许多图书可供挑选,也可以完全像许多编程人员那样,直接查阅官方的 Lua 参考手册。

《Network World》主编谈“Lua 编程语言” II

现在,假设你拥有 iPad,可能想要试一下 Codea for iPad,这个编程工具由名称很古怪的 Two Lives Left 开发商发布,它把 Lua 解释器嵌入到了为 iPad 设计的 iOS 应用程序中。

这是个外观漂亮的程序,有优美的图形和设计精良的用户界面。

从起始屏幕,你可以入一个现有的项目,也可以新建一个项目。Codea 项目包括:分页窗口(构成应用程序的类在不同的页上)和执行时候的左右两栏布局,上面有程序参数面板、文本输出面板、左边的执行控件(后退、开始、暂停和重置)以及右边的程序图形显示面板。

Codea 编程环境真正吸引人的地方是,它与 iPad 集成得非常好,简直是浑然一体。如果你想更改数字,只要用手指戳一下数字,然后放开,用键盘来编辑,也可以按住,同时上下拖动,即可增加或减小数值。想更改颜色,只要用手指戳一下值,就会弹出颜色选择器。你还可以选择精灵(sprite),操作方式几乎一样。还内置了 Lua 的完整说明文档。

演示程序给人留下了很深刻的印象。一款演示程序模拟了你在移动 iPad 时,一块布很自然地摆布(Codea 系统能够访问来自方向感应器的数据,并同时检测多个屏幕触摸动作),有几个多点触控演示、几个场面相当复杂的游戏以及鼓机。

缺少了什么功能?缺少了在本地或与外部资源之间导入或导出代码的功能,缺少了访问网络功能的功能,无法隐藏左侧的控制面板,以便应用程序可以全屏运行,还缺少了各种代码保护功能,所以其实无法编写实际用在贵企业的应用程序。

Codea 开发者 Simeon Nasilowski 告诉我,他打算进行各种各样的改进,很可能包括我提出的所有需要改进的方面;但眼下,他还没有答应给出时间表。

Simeon 告诉我,自 Codea 在 10 月 26 日发布以来,大约已售出了约 10000 套,每套售价 7.99 美元,这个成绩相当骄人。Codea 在满分是 5 分的评价中得到了 4 分。



作者简介:30 多年来,作为《Network World》的主编马克·吉布斯发表了大量 IT 产业的文章。同时,马克还是负责英国 Novell 公司技术服务管理运作的联合创始人。

51CTO 原文地址

<http://mobile.51cto.com/hot-302363.htm>

■ 编者按

为什么一款 N 年前的游戏还能坐在排行榜的前列,是玩家百玩不腻么? 我看未必。

手机不如家用机 移动游戏为何总倚老卖老?

刚看了一篇 Windows Phone 的图表报道,突然郁闷了。

玻璃渣(Blizzard 暴雪)的粉丝肯定绕不了我。

当然,小编这里不是在否定一些移动新游的

Highest ranked paid applications in Windows Phone Marketplace | United States | October 2011 only

RANK	APPLICATION	CATEGORY	PRICE
1	Angry Birds Microsoft Studios	Games Action & Adventure	\$2.99
2	Burn The Rope Microsoft Studios	Games Puzzle & Trivia	\$2.99
3	Gravity Guy MindUp	Games Action & Adventure	\$2.99
4	Implode Microsoft Studios	Games Puzzle & Trivia	\$2.99
5	COLLAPSE GameHouse Live	Games Puzzle & Trivia	\$2.99

Source: Daino

图中的意思是,《愤怒的小鸟》这款源于 iPhone 的游戏继续依旧占据 Windows Phone (以下简称: WP) Marketplace 平台的头名,第二名是《切绳子》。

其实不单是 Marketplace,《切水果》在谷歌 Android market 里面也拍在了第一位,排名靠前的自然还是有《小鸟》等老面孔……诸如此类的平台很多,小编就不一一举例了。

为什么一款 N 年前的游戏还能坐在排行榜的前列,是玩家百玩不腻么? 我看未必。是游戏更新换代很频繁,创意抢在了玩家需求前头,如《里约版小鸟》? 真正玩了之后,发现这款游戏也就哪么回事。小编作为一个老玩家,有点看不明白了。

新的移动大作在哪里?

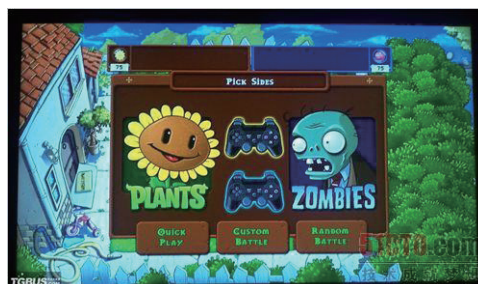
看了这些“老将”依旧常年驰骋在移动应用的,我在想新的精品游戏在哪? 难道《小鸟》和《切水果》是移动领域的 WOW 么?! 如此相比,

成长速度,有些优秀的新游 APP 排名也已非常靠前,这就说明了它的成功。但只是感觉这类水准的游戏还是少了一些,或者是没有被消费者所注意到。

我们 51CTO 在之前已经报道过许多如何使开发者在应用商店盈利,如何改善自己应用排行排名的技术文章,但似乎还没能起到撼动这些“老将”头名交椅的作用。

凑巧的是,小编刚刚在网上无意中看到了索尼 PS3 推出《植物大战僵尸》的新闻,带着鄙视的心态点进去看个究竟。发觉 PS3 版的《僵尸》和移动版的真的大不一样。

家用机版《植物大战僵尸》可对打



手机不如家用机 移动游戏为何总倚老卖老？ II

看到这幅图后,小编震惊了,这货还是《植物大战僵尸》么?



这个是移动和 PC 版本上不曾有的模式——双人对打,支持单机双手柄同屏游戏,僵尸方面也是收集类似阳光的“大脑”用来生产僵尸。除此之外,双人模式还支持两位玩家都是植物,每人可以选择 4 种植物,共同作战。天上掉下的阳光需要两人同时碰触才可以收集……

震惊之余,小编赶紧搜索了下其他家用机版本的《植物大战僵尸》,才发现 2010 年 10 月微软 Xbox 就有了这款双打产品, PS3 这款产品只不过是移植之作。

对于一款已经放弃很久,对其玩法已经烂熟于心的一款游戏,小编在此勾起了浓厚的兴趣。家用机平台可以把这款单机老游戏提升到了对战的高度,我很喜欢。相信身边同岁的朋友很少有不会玩这个游戏的,都觉着自己不含糊的话终于可以有个平台来较量一下了。

其实仔细想想也对,一款人们玩了好几年的游戏,搬到了另一个平台(Xbox 或 PS3),如果不能拿出令人信服的新作,玩家会买账么? 写到这里,笔者又想起之前说的各移动 APP 平台的大佬们,哎。

结语

小编本想说移动平台的应用后续开发力度

不够,缺乏创新,但被瞬间想起的资讯所击倒——《Android 和 iOS 游戏收入将超任天堂索尼》。

移动数据分析机构 Flurry 的数据显示,美国便携式游戏软件市场, Android 和 iOS 游戏软件所创造的收入将超过任天堂和索尼掌上游戏设备的收入总和。

这说明什么,说明人家发展的很好。后续开发力度并不大的移动产品已经令某些开发团队、某写应用平台赚个锅满盆盈,但对消费者来说并没有什么意义。

消费者并没有获得新的游戏快感,也许就会放弃某些平台产品。如果是那样的话,对开发者来说,前景并不乐观。



画外音: 小鸟愤怒了,玩家早愤怒了,有人管么?

PS: 听说 NFC 手机要推出对战版本的《愤怒的小鸟》,小编对此很是关注。

但悲剧的是, NFC 的手机普及率少的可怜,弄个新版本很难么,移动游戏开发团队咱不糊弄消费者们,咱做的东西别倚老卖老了,行么? ■

51CTO 原文地址

<http://mobile.51cto.com/AppFocus-304236.htm>

■ 编者按

随着移动互联网操作系统领域“三足鼎立”趋势渐成,越来越多的开发者准备加入到 Windows Phone 的开发大军中来。不过在此之前还有一些问题需要提前搞清楚:比如,Windows Phone 开发的市场前景如何?Windows Phone 平台与其他移动操作系统有什么不同?有哪些细节需要注意?

马宁: 越早进入 Windows Phone 开发越有优势

针对以上这些问题,51CTO 独家采访了 OpenXlive 的首席技术官马宁先生。马宁先生曾任前微软技术合作经理、亚洲工程院软件工程师,从事 Windows Phone、Windows Mobile 和 Windows Embedded Compact 技术推广和开发工作,对微软技术和 Windows Phone 相关开发事项非常熟悉。

涌现了很多个人开发者,但等到积累到 50 万个应用以后,一些小的团队,哪怕你的创意再好也很难冲到排行榜前面去,“现在 Windows Phone 只有三万六千款应用,只要你的创意够好,设计够精美,很容易冲到前 20 或前 10 的排名上去。这样让小团队增加他们的曝光率,小团队仍然能保持一个优势”。据马宁介绍,目前 Windows Phone



提早进入 Windows Phone 开发将获得先发优势

马宁认为,目前开发者投入 Windows Phone 时机恰好,因为 Windows Phone 平台正蓄势待发,市场前景广阔而竞争又没有特别激烈,个人开发者和小团队比较容易在此平台上脱颖而出。“往往早期进入一个市场的话开发者会获取以前没有的优势”,马宁举例说,苹果 App Store 早期的时候

开发者两个比较大的来源,一个是传统的 iOS 及 Android 开发者,另外是以前 .Net 转过来的,之前做微软这条产品线的开发者。根据 OpenXlive 的数据统计,目前绝大部分 Windows Phone 应用程序都是 .Net 开发者提供的,只有 10% 左右是 iOS 及 Android 开发者提供的。马宁认为在半年以内这一分布还将持续,随后可能其他平台转移过来的开发者数量才会慢慢上升。

马宁：越早进入 Windows Phone 开发越有优势 II

Windows Phone 平台：封闭的好与坏

相比 iOS 和 Android 系统，Windows Phone 最直观的改变就是 Metro UI 的运用。用马宁的话说，苹果“重新定义”了智能手机的 UI，但随后由于几乎所有智能手机都是采用了类似的设计，“文艺青年慢慢变成普通青年了”，Metro UI 的出现打破了这种同质化现状：“Metro UI 第一次将图形化的视觉元素与应用内容整合到一起”，马宁说。Metro UI 里应用的图标比较大，因为它不止要显示这个应用是做什么的，还会显示一些缩略信息。“Metro UI 与 Apple 的 UI 走的是不同的路，Apple UI 是绚丽之极，它几乎把所有的细节都描述的非常好，Metro UI 几乎没有任何自己的风格，但是如果你加一些背景图片，加一些自己的元素进去，你会发现这种 UI 是可以适用任何的风格的”。

将所有应用的缩略信息都提供在首页，会不会造成系统运行速度降低？马宁表示，其团队成员曾做过测试，最多把 100 个图标放在 Windows Phone 首页，没有出现影响系统性能的情况。笔者也曾经试用过几款 Windows Phone 手机，发现在类似配置的情况下，Windows Phone 手机的运行速度比其他系统要快一些，这是为什么呢？

马宁称，这是由于 Windows Phone 系统相对封闭，应用调用系统级程序受到严格的限制，同时 Windows Phone 后台最多只支持 5 个任务，它会优先把 CPU 的运算能力提供给前台的运算程序，同时限制后台程序的 CPU 占用量不能超过 10%，因此能保证前台的应用的性能非常平滑。

封闭性为 Windows Phone 带来一些性能优

势的同时也让一些开发者望而却步。一些从其他平台转过来的开发者可能不太适应 Windows Phone 的诸多限制。但马宁认为，从长远来看这种封闭机制能够保证整个操作系统安全平稳运行，未来 Windows Phone 可能会变成三个移动平台 (iOS, Android, Windows Phone) 里最安全的一个平台，这对直接用户是有利的。“对于开发者来说可能要面临这样的取舍：我进入这个平台可能要抛弃一些相应的东西。但是将来 Windows Phone 发展的比较好的时候，他获得的回报会比他付出的多”，马宁说。

游戏类应用有很大发展空间

从应用种类上来说，目前各类移动互联网应用平台中，游戏娱乐类应用往往占据比较大的比例。马宁称，Windows Phone 平台现在仍有大量的非游戏类应用的空白需要填补，预测未来 6 个月内仍然会以这些应用作为主要的发展趋势，当这类应用程序满足到一定程度，比如 10 万或 20 万，用户群会被无限细分的，这时候再开发同类应用就很难获得足够大的空间了。但是游戏类应用不同，其开发空间是非常大的，而且不会受到时间的限制，也因此，游戏是开发者一个很好切入的方向。不过，马宁建议开发者做应用还是要选择自己最擅长的领域：“手机也好 PC 也好，都是对于需求的展现，从 PC 切换到移动设备上，有很多东西可以互相借鉴的。不要贸然的去尝试一些陌生的领域”。

Windows Phone 上专门针对游戏开发者提供了服务平台 Xboxlive，但由于该平台是采用邀请制的，约有 95% 的游戏开发者无法享受到该服务，

马宁：越早进入 Windows Phone 开发越有优势 III

以马宁为首的 OpenXlive 团队便开发了一个第三方平台为这些游戏开发者提供相关的开发和推广服务。OpenXlive 提供的服务包括积分榜、应用成就、游戏调查推广、数据查询分析,以及 SNS 功能等。这一领域目前还没有竞争者,不过未来随着 Windows Phone 平台的发展壮大一定不会出现更多类似服务。“我们希望自己将来在第三方游戏服务产业里面进入第一梯队。我们一直在从其他平台上学习更多的技术,我们也关注 win8, .Net 的技术发展,我们希望能把自己在微软技术上的优势,转换成为开发者提供的服务集成的平台”,马宁说。

决方案。“我们跟微软的产品经理有过交流,微软未来能肯定支持其他支付方式的,但是需要花一些时间,所以在此之前我们也是希望能与国内的一些第三方支付平台有一些合作。尽快的让开发者享受到服务”,马宁表示。

尽管目前一些开发者在尝试免费提供应用,然后通过广告等其他方式获利,但马宁认为这种方式不会成为微软 Windows Phone 平台上的主流盈利模式。“微软整个 Market Place 的设计还是希望能够通过收费的方式促进应用开发者获利。这样开发者才有更多热情去做开发。”马宁说,“现在是有些问题,比如国内的 Market Place 访问



Windows Phone 的支付瓶颈

困扰 Windows Phone 开发者的最大困难是应用支付问题。Windows Phone 不支持信用卡绑定外的其他支付方式,这对一些开发者获利形成了障碍。马宁表示,支付是西方国家和亚洲地区国家最大不同的地方。信用卡绑定这种消费方式在国外没有问题,但中国用户普遍喜欢先免费下载应用,然后需要一些增值服务的时候再去支付一定的费用,这就要求微软提供一些本土化地解

不正常,下载失败,还有其他的一些技术性的因素,比如支付的渠道没有打通,但这只是一个暂时的现象,等到 Market Place 真正在中国落地以后,所有这些问题都能得到相应的解决,届时开发者就能从平台上赚到钱,整个生态系统就能更好的运转下去”。

51CTO 原文地址

<http://mobile.51cto.com/news-302775.htm>

■ 编者按

熟悉 .NET 的朋友一定听说过《你必须知道的 .NET》这本书,这本书的作者王涛先生专注于 .NET 底层架构和 .NET 平台企业级应用,长期投入于互联网产品开发、微软云计算平台和企业级系统架构的研究与实践,连续三届微软 MVP。最近,王涛的履历表上又多了两个新的身份:Windows Phone 开发者和移动互联网创业者。

王涛:你必须知道的Windows Phone

创业:移动互联网试错成本低

在回答为何选择移动互联网和 Windows Phone 作为创业方向的时候,王涛向 51CTO 记者表示,由于中国手机用户数远远高于 PC 用户,未来无论是从用户群体还是移动业务上,移动互联网产业都会有一个大的发展,而且“创业要顺应潮流”,目前移动互联网风头正劲,“抓住机会有可能做出一些事情来”。确立了这一方向后,作为一个 .NET 开发者,选择将 Windows Phone 切入点则顺理成章。



今年 8 月份,王涛和几个朋友离职创业,开始做他们的第一款 Windows Phone 产品“海纳”。最初的设想是做一款个性化阅读类的应用,因为“随着 web2.0 的概念出来后,人们获取信息的方式变成推荐机制,不是我去找信息,而是信息找我”。但是,经过一段时间的研发后,他们发现,对于只有五六个人的“海纳”团队来说,要应付海量的内容有些吃力,于是他们及时做了调整,将个性化阅读改为个性化应用推荐。两款产品后端的推荐机制基本是相似的,所以转型并没有带来很大困难。目前,定位于个性化应用推荐的“海纳”第一版已经上传到微软 Market Place,正在审核中。

“移动互联网试错的机会更容易”,王涛说,他表示相比互联网产品,移动互联网产品不论是从开发周期还是产品定位上都要“轻”一些,一次尝试不成功,还可以很快转到另一个。另外,由于产品从开发到投入市场的周期相对较短,你可以很快看到用户的反应,“这对开发者来说是件很有意思的事情”。

基于云计算的移动应用

尽管从“个性化阅读”转型为了“个性化应用推荐”,海纳要处理的数据量仍然相当庞大。

“后端推荐服务计算量非常高”,王涛举例说,“假设你有 10 万个用户,10 万个应用,要计算出

王涛：你必须知道的 Windows Phone II

用户最喜欢的应用,那就是 10 万乘 10 万的运算量级”。对于小团队来说,这样的运算量级自己搭建服务器成本是很高的,此外,移动应用通常有比较大的爆发力,一款好的应用其用户量的增长经常是几何级的,普通服务器无法应对这样的动态需求增长,常常令开发团队在用户量突增时“悲喜交加”。基于对海量数据运算能力和弹性空间的要求,“海纳”选用了微软 Windows Azure 云计算平台承载其后端服务。“不过,我们不是所有服务都放在云端”王涛说,“只有计算量高的放在 Azure 上”。

那么,开发基于云计算的移动应用和普通应用有什么不同?

王涛介绍说,开发云端应用要考虑很多问题,比如 performance 的问题,你不能把数据一次性 POST 到云上,而要考虑异步的操作,小数据多次往上放。此外还有安全类问题,比如敏感数据的处理。用户操作习惯问题:哪些数据放在云上,哪些不放在云上,放在云上的是否要在本地有缓存等等,都要慢慢去了解。但从开发技术本身来说差别不大,现在关于云上服务都提供了很好的 SDK,你不需要关心太多技术细节。王涛提示,云端应用在测试时比一般应用稍微麻烦一些,但对开发周期影响不大。

谈到移动应用与云计算结合的未来趋势,王涛表示苹果的 iCloud 算是一个先驱,它让手机的应用自动同步到苹果的云中心,用户不再需要关心数据的存储和过多技术上的东西。王涛预测未来会有越来越多的第三方应用会同步在 iCloud 上,但遗憾的是 iCloud 支持苹果平台,他

希望未来会有支持所有平台的云计算服务。以马宁为首的 OpenXlive 团队便开发了一个第三方平台为这些游戏开发者提供相关的开发和推广服务。OpenXlive 提供的服务包括积分榜、应用成就、游戏调查推广、数据查询分析,以及 SNS 功能等。这一领域目前还没有竞争者,不过未来随着 Windows Phone 平台的发展壮大一定会出现更多类似服务。“我们希望自己将来在第三方游戏服务产业里面进入第一梯队。我们一直在从其他平台上学习更多的技术,我们也关注 win8, .Net 的技术发展,我们希望能把自己在微软技术上的优势,转换成为开发者提供的服务集成的平台”,马宁说。

Windows Phone 的支付瓶颈

困扰 Windows Phone 开发者的最大困难是应用支付问题。Windows Phone 不支持信用卡绑定外的其他支付方式,这对一些开发者获利形成了障碍。马宁表示,支付是西方国家和亚洲地区国家最大不同的地方。信用卡绑定这种消费方式在国外没有问题,但中国用户普遍喜欢先免费下载应用,然后需要一些增值服务的时候再去支付一定的费用,这就要求微软提供一些本土化地解决方案。“我们跟微软的产品经理有过交流,微软未来能肯定支持其他支付方式的,但是需要花一些时间,所以在此之前我们也是希望能与国内的一些第三方支付平台有一些合作。尽快的让开发者享受到服务”,马宁表示。

尽管目前一些开发者在尝试免费提供应用,然后通过广告等其他方式获利,但马宁认为这种方式不会成为微软 Windows Phone 平台上的主

王涛：你必须知道的 Windows Phone III

流盈利模式。“微软整个 Market Place 的设计还是希望能够通过收费的方式促进应用开发者获利。这样开发者才有更多热情去做开发。”马宁说，“现在是有些问题，比如国内的 Market Place 访问不正常，下载失败，还有其他的一些技术性的因素，比如支付的渠道没有打通，但这只是一个暂时的现象，等到 Market Place 真正在中国落地以后，所有这些问题都能得到相应的解决，届时开发者就能从平台上赚到钱，整个生态系统就能更好的运转下去”。

你必须知道的 Windows Phone

《你必须知道的 .NET》一书以通俗易懂，细致全面的方式向开发者描述了 .NET 开发的一些细节和知识点，受到了读者的广泛好评。那如果再写一本《你必须知道的 Windows Phone》都会包括哪些章节呢？

面对 51CTO 记者这个问题，王涛略作思考后表示，如果真要写这样一本书，他会在第一章首先强调手机上应用与桌面及 Web 应用的差别。他表示移动开发与 PC 上的开发有很大不同。“更多不是开发技术的问题，而是你对移动产品的思路转变问题”。比如 PC 应用可以让用户大量的键盘输入，但在手机上就不行；比如按钮的摆放位置你要考虑用户是左手操作还是右手操作；手机联网你要考虑什么时候联什么时候不联，是否设置离线功能等等，很多细节的设计其实首先源于思路的转变。对于从传统领域转移过来的开发者来说，这点尤其重要。王涛说，之前的 Windows Mobile 之所以失败，就是因为它是用 PC 的理念去理解手机的操作系统。

具体到 Windows Phone 开发，王涛强调基础是最重要的。Windows Phone 开发者要打好两个基础，一个是语言基础，一个是运行时基础。所以书的第二部分会是一些关于这两方面的开发基础和细节的介绍。

第三部分是 Windows Phone 最佳实践。王涛说，“涉及具体开发和产品的书，一定要强调实践中的经验”。学会从其他产品中吸取经验教训也是开发者必须学会的技能。“海纳”在开发过程中，也研究了很多同类的产品，包括 Android 和 iOS 上的。王涛表示，现在 Market Place 上已经有四万款应用了，但是相比 Android 和 iOS 上的应用 Windows Phone 的应用大多还不够精致，所以“现在是追求质的时候了”。

在书的最后，王涛表示，他会做一些移动应用与云端技术结合的探讨。

我们不知道王涛先生是否真的会写这样一本书，问这个问题的初衷是，希望他能为广大准备加入 Windows Phone 开发的朋友们梳理出一条学习的脉络。根据 51CTO 的调查，目前 Windows Phone 开发方面的书籍资料确实不够多，特别是关于“最佳实践”方面的分析介绍非常欠缺，这也是因为 Windows Phone 开发还刚刚起步，值得推荐的原生应用实在乏善可陈——这是 Windows Phone 目前的弱势，却也正是开发者们的机会。抓住机遇，也许下一个红遍全球的移动应用的作者就是你。



51CTO 原文地址

<http://mobile.51cto.com/news-304997.htm>

■ 编者按

随着诺基亚 Windows Phone 手机的推出,国内开发者对 Windows Phone 平台开发的关注度又掀起了一股热潮。这次我们请到了 Symbio 公司的 Windows Phone 开发专家李靖南老师,为大家谈一谈 Windows Phone 应用开发中的一些细节。

李靖南：详解 Windows Phone 开发细节及注意事项



在今年十月份的微软 TechED 大会上,李靖南老师曾为大家带来了《开发 Windows Phone 多媒体应用》和《利用 Expression Blend 设计不一样的 Windows Phone 7 应用程序》两门课程,实际上,李靖南老师不仅对微软技术非常熟悉,他本人开发的一款叫做“市民日记”的 Windows Phone 应用在微软 Market Place 上已经取得了不俗的排名,并且在发布了简体中文、繁体中文、英文、意大利文、韩语版后,仍在陆续开发新的语言版本。在采访中,李靖南从一个开发者的角度详细描述了

Windows Phone 应用开发的一些细节优势以及注意点。

Expression Blend 拖拖拽拽的可视化组件

谈及 Windows Phone 开发与 iOS、Android 等平台开发的不同,李靖南表示,基于 Xbox360 和 PC 上的游戏经验,微软 XNA 比 iOS 开发游戏要快捷很多,此外, Silverlight 采用 XAML 这一描述性语言,这点与 Android 非常类似。在开发工具方面 Windows Phone 也基本都有与 iOS、Android 一一对应的工具,所以开发者不管是从 iOS 还是 Android 上切过来都会相对容易。

李靖南称,微软在 API 接口方面做了很多优化,提供了一些比较强大的开发工具,很多功能都可以采用“拖拖拽拽”的方式实现。其中 Expression blend 就是很有效的一个工具。

在 TechED 的 Windows Phone 相关课程里, Expression blend 课程排名第一,这也从一个侧面说明了这一工具的受欢迎程度。

“2008 年开始用 Expression blend 的时候,我也曾陷入一种迷惑:这个东西到底是给开发者用的还是给设计者用的?现在我觉得它还是多少植入了跟开发相关的知识细节在里面,比如数据绑定,还有开发模板化等,这些概念一个纯粹的设计者理解起来还是有难度的”,李靖南说。所以,他

李靖南：详解 Windows Phone 开发细节及注意事项 II

认为 Expression blend 更适合“有一些设计感的开发者”和“有一些代码基础的设计师”——李靖南称之为“桥梁工程师”的群体使用。一方面，项目团队可以通过这些“桥梁工程师”进行很好的沟通，并可在项目初始阶段用 Expression blend 做一些可实现基本交互功能的应用原型演示；另一方面，具备“桥梁工程师”特质的开发者或设计者可以独立用 Expression blend 开发一些简单的应用，比如对界面美观度要求没那么严格的商务类应用，或者对程序需求要求不那么多的交互类应用。

基于 Expression blend 这种工具，传统意义上的开发者与设计师的关系得以重新定义：“做网页版开发的时候开发与设计角色分的非常明确，但在移动开发项目中，我们建议开发人员也要向设计渗透一些，设计人员也要多少了解一些开发有关的事情，当一个设计师懂得一个东西工作的原理之后对他来说设计一个东西更有利”。

要善用 Windows Phone 开发控件

在 CodePlex 开源平台上，微软为 Windows Phone 开发平台提供了名为 silverlight toolkit for windows Phone 的开发工具，其中包括很多模块化的控件，这些控件目前有几十种，基本能满足普通的开发需求。李靖南建议开发者在设计应用之前，先把这些控件都下载到手机里一个一个研究一下，了解基本情况，这样不但可以拓宽你开发的思路，同时，也避免你在一些已有的功能开发上花费太多的时间。

在采访中，李靖南重点向 51CTO 记者介绍了两款控件，分别是 Panorama 和 pivot。这两款

都是界面布局控件，其中 Panorama 是目前最受欢迎的工具，它支持背景前景不同步的滑动，使得应用有一种纵深关系的背景图。“Panorama 特别适合做时尚类的、新闻类的界面”，李靖南说。

除了微软官方提供的这些控件，一些以前在 silverlight 开发领域比较强势的公司也都有针对 Windows Phone 的第三方控件发布，不过这些控件大多是收费的。如果开发者有需求并且有预算的话也不妨考虑用这些空间。唯一遗憾的，目前并没有一个统一的平台收集所有的控件，这些第三方的控件大都发布在自家的宣传平台上，不过李靖南分享了几个他常去的 Windows Phone 开发相关网站，供开发者参考：silverlightcream.com 是必看级别的，由一个微软的工程师维护，他几乎收集了所有值得推荐的与 silverlight 有关的技术文章；另外关注微软官方的 channel9 频道可以获取一些最前沿的技术资讯。国内网站的话，并没有专业级别比较高的，李靖南推荐的是 silverlight 中国，cnblog，当然，还有 51CTO。

设计应用时应该注意的细节问题

Windows Phone 手机在硬件上设置了三个常用键，左 BACK 键、中间 HOME 键、右 SEARCH 键。李靖南表示，开发者在设计应用时要多考虑与这些硬件有关的操作细节。

首先，理论上 BACK 键可以无限制回退，但是用户往往不记得自己上一步操作或者前几步操作进行到哪里了，所以应用设计不要过多的依赖 BACK 键，比如可以考虑自己设置一些图形化的、美观的指引，提醒用户目前的位置，好让他知道往前或往后可以去到哪里。

李靖南：详解 Windows Phone 开发细节及注意事项 III

其次,最右边的 SEARCH 键设置的是直接切入 Bing 搜索界面,这一设置显然有很浓的商业味道在里面,但是用户尤其是国内用户使用 Bing 搜索的需求其实并不是很大,而这一设置会在一些情况下给用户造成一些困扰。“比如在打游戏的时候,不小心就碰到了,然后你就会被从游戏界面切出去到 Bing,这是很不好的用户体验”,李靖南说,他对 Windows Phone 的硬件设置持保留意见,但是作为开发者,只能在既有基础上考虑解决方案。

此外,应用设计还要注意界面的统一性。比如“市民日记”采用的是亮色背景,而 Windows Phone 默认的弹出消息对话框背景色取决于手机本身的设置,常常显示为黑色,因此,如果使用默认功能,就会有点不和谐。另一个需要开发者考虑的问题是,应用的数据如何做持久化,是保留到本地?还是采用云服务?“这是你事先要考虑的问题”,李靖南说。

缺乏原创的杀手级应用

Windows Phone 相对封闭开发制度在提高系统安全性和运行速度的同时也给开发者带来了一些困扰。李靖南说“作为开发者而言我或多或少感到了受束缚的感觉”。曾经有一次一个客户拿着安卓手机问李靖南能不能做到类似锁屏界面快捷方式的功能,由于 Windows Phone 对这一系统功能的调用限制,李靖南只能遗憾的摇摇头。“这就像硬币的两面”,李靖南说,开发者需要在这其中做一些取舍。“跟桌面应用开发最大的不同是,手机资源受限制的,桌面时代是不断的往里加,不断的加一些功能,做的大而全,手机上反而是看有

什么不必要的放下去,在有限的界面区里展示更重要的内容”。

那么目前在微软的 Marketplace 上有什么比较有价值的 Windows Phone 应用呢?谈到这个问题,李靖南表示非常“尴尬”：“这对做 Windows Phone 开发是不大不小的一个刺激”,李靖南经常拿着自己的 Windows Phone 手机向亲朋好友“推销”这个产品,“但是我发现我所能推销出来的一个代表性的、在 Windows Phone 上独有的、首创的应用,目前来说基本上没有”。遥想一两年前,拿着 iPhone 的人可以很自豪的向你介绍愤怒的小鸟,但在 Windows Phone 7 上,现在还没有这样级别的应用。“游戏方面一直是 Windows Phone 的强势,因为他本身有 XNA 这样的平台支撑。在这块多多少少有一些突破。Xbox360 上一些很有名的游戏简化版本在 Windows Phone 上都有。这些反而是 iOS 和 Android 上没有的,因为他缺少游戏机级别的支撑”,李靖南说“但是游戏之外的,还有很多时尚类,儿童教育类,理财等这方面我希望看到更多 Windows Phone7 上的亮点”。

最后,李靖南总结道:“我特别想呼吁大家,如果有个好的 idea 一定要先放到 Windows Phone 7 上去,这样对两方面都有好处。对开发者个人来说,由于这是崭新的平台,并不像有几十万应用的 iPhone 平台上竞争那么激烈,Windows Phone 7 上你只要做的优秀就能够保证一两个月内就让全世界人都看得到。我认为至少所有 .Net 的开发人员一定要尝试一下”。■

51CTO 原文地址

<http://mobile.51cto.com/windows-phone-303400.htm>

51CTO 专项调研:2011年度移动开发者生存状况

iPhone 手机的出现,引领了智能手机市场的迅速崛起,基于 iOS、Android 等系统的移动应用开发方兴未艾,引的众多开发者投身于移动开发领域。再加之 APP Store 和 Android 等应用商店,日渐完善的合作流程与便利的支付方式降低了移动开发创业门槛,更有一大批开发者独自或几个志同道合的技术人员开始了创业之路。

面对移动开发这样一个新兴行业,引起了 IT 界高度的重视和广泛的关注。国内外各大 IT 技术公司、技术网站或论坛等针对移动开发行业方方面面展开了很多权威调查,让我们对移动开发领域有了一个越来越清晰的认识。毕竟,IT 技术行业日新月异,每天都在上演着行业的剧变。作为 IT 技术的推动者、行业弄潮儿的移动开发者生存状况如何呢,是否与外界所看到的一样光鲜?还是在光鲜外表下,也有说不出的痛呢?

生存状况的研究可以有多个方面,例如收入、健康、休闲、发展等等,收入是主要的评估指标,也是较为普遍的用以衡量行业从业者工作情况的晴雨表。影响收入的因素有很多,例如工作城市、所在公司的规模、开发者个人能力等等,只有通过多个因素的分析,得出的结果才具有参考价值,因为即使同属一个行业,收入差距也很大。

移动开发者整体收入情况

51CTO 本次调查发现,整体的收入分布呈现正态分布,46% 的开发者收入处于 4001 元到 8000 元之间。根据国家统计局公布 2010 年城镇私营单位和非私营单位年平均工资数据显示,全国信息传输、计算机服务和软件业的平均年薪为 66598 元。

移动开发者掌握的开发语言情况

Android 开发促使 Java 程序员需求增加,对比最新的编程语言排行榜我们也可以看到移动开发对整个开发语言环境的影响。Android 平台底层是由 C 语言开发的,加之 Android 的开源政策,iOS 和 Windows Phone 平台迅速发展,需要大量 C 语言系列的程序员。

城市间收入水平差异情况

51CTO 调研发现,北京、上海、广州、深圳作为中国的一线城市,移动开发者的平均收入水平高于其他城市。8000 元以上收入北上广深达到 25.4%,其他城市为 8.0%,尤其是万元以上收入人群,北上广深达到 12.7%,其他城市只有 4.3%。

除以上内容外,此次报告还包括:移动开发行业基本情况分析:移动开发者对于开发平台选择情况分析、移动开发者开发应用情况分析、移动开发需要的技能情况分析;影响收入情况的因素分析:工作地域对收入情况的影响分析、公司性质对收入情况的影响分析、公司规模对收入情况的影响分析、工作年限对收入情况的影响分析、工作强度对收入情况的影响分析、工作内容对收入情况的影响分析等多个内容板块。

希望本次调研能够帮助您更多的了解移动开发行业,了解移动开发者真实的生存状况。下载地址: <http://down.51cto.com/data/282609> ■